

EPISODIO 3

350

Richard mandó a Gilfoyle montar un servicio oculto que mantuviera a flote "El Flautista" pero este ya no recuerda donde se encuentra. Gracias a dios, como buen sysadmin, siempre hace backup de todo su trabajo, pero se trata de backups un tanto peculiares... Gilfoyle guarda el trabajo que hace en archivos encriptados relacionados con temáticas que le gustan.

Tenemos el fichero que contiene información sobre el servicio. Necesitamos que extraigas la información, accedas al servicio y consigas la flag de UAM. ¡Mucha suerte!

Enlace de descarga: <https://drive.google.com/open?id=1qTul9VndJ24krrO8U1WF3JpS77M4M2hV>

Info: La flag tiene el formato UAM{md5}

Descargamos el fichero que nos proponen, Marvel.zip de 30 MB un merecido homenaje al recientemente fallecido Stan Lee.

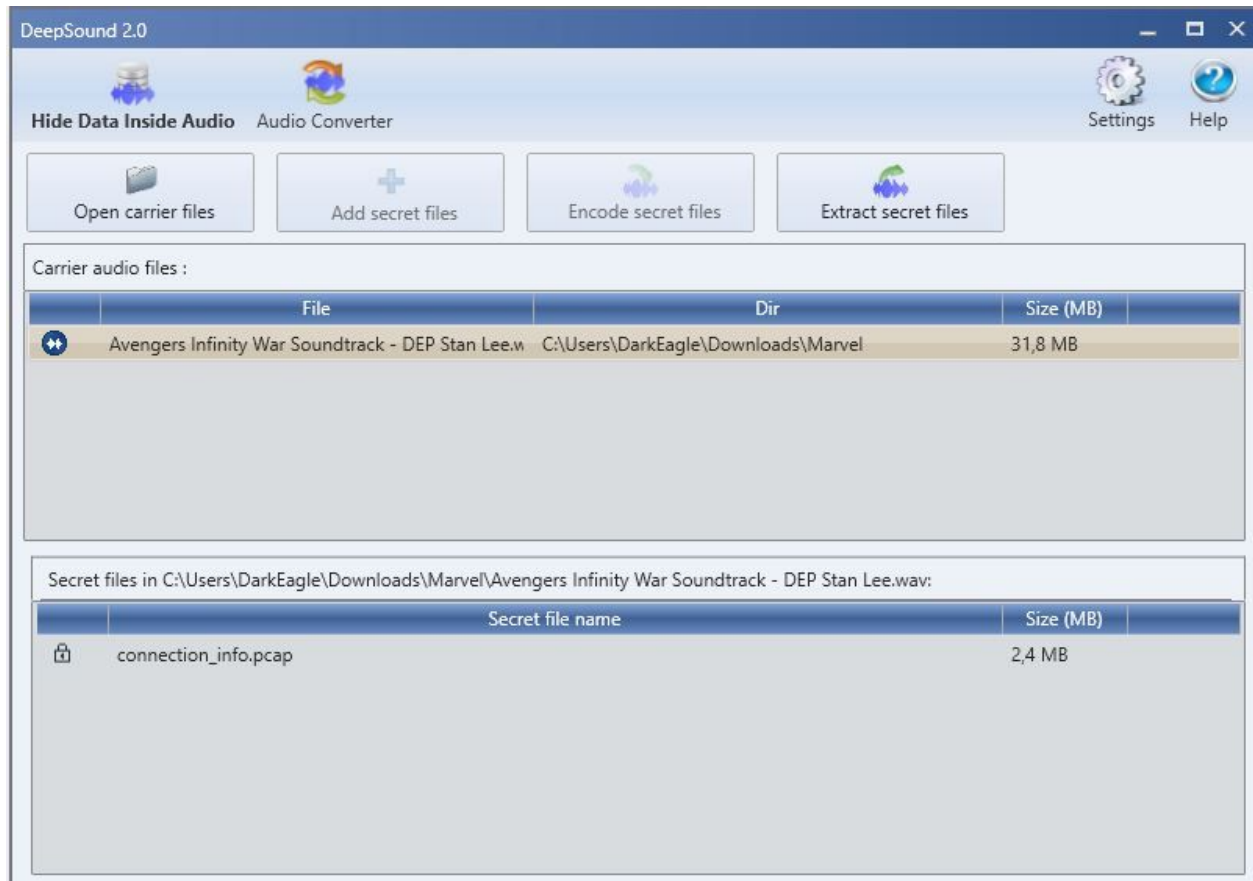
Al descomprimir el fichero encontramos un wav, vemos los strings, y en especial la última línea:

```
root@kali:~/unaalmes# strings Avengers\ Infinity\ War\ Soundtrack\ -\ DEP\ Stan\ Lee.wav
| tail -1
U29uaWRvUHJvZnVuZG87KQo=
```

Lo decodificamos de base64:

```
root@kali:~/unaalmes# strings Avengers\ Infinity\ War\ Soundtrack\ -\ DEP\ Stan\ Lee.wav
| tail -1 | base64 -d
SonidoProfundo;)
```

Ok, tenemos una pista .. probamos las típicas cosas de stego en audio .. Sonic Visualizer, Audacity .. y es usando DeepSound (Sonido Profundo) donde tenemos un resultado.



Extraemos el fichero y lo analizamos con Wireshark.

Analizando las tramas TCP en el stream 35 vemos algo curioso:



Pasamos el morse a texto:

Input data

Convert

morse_code to text

Output:

34punto247punto69punto86dospuntos1337

```
root@kali:~/unaalmes# nc 34.247.69.86 1337
```

```
C0Y{0@0n70`俱V0Q0m@d-0~0u70aC@70<c>8GfG,QQQ @I0000#00080506#00b09Eo~-0D00#J;00S@Q000Q000
0J000000-Z00/gn0L003[0c000l0s0->0I0\000H00XPb0y00iK0;K00J000=zX000HC000>0 00w0f          0Vp
                                                    1f00N000#0}0
                                                    E0+C0S0f'='
0:0-拆0000v0|0<00Hz0V000
[=0'00'!003:00z0!R00ma0eN70J0#B/08N0210^0%v0aat0701T0++0000n<00?R0T0"0l0.060M;0
0S00DPx0+/'Y000SBn00u{50p010eI(c000FX00900s0X@8N00V00pf``0~0001fP      A0<g0??0Y!000000-0I7u06)00}[G
0000C0E64]0-0lK00
0000B0hU00G00[0-T0`0k0/000
0x00000@00Y000000D00W0W0000000pl
    =0:7#000#000*e5[F00[00c90)0000E0(0|H0t00000P'+00200#00%
                                                    z0
00S090bR0=7100VMPO      0; 0Y0$XK000\Ru000q0|000049+Jhu000S00T0      k00000000_0J!4{B0QS000*z;c00?0
!!i+0[0]Exy000000!pEMo6000000000[R0e0000000
        06
000(00000000)-00,R0Wy000/'ot0000Of0400K0000<%N<00r00sYD0D0`0100o6%30-J00S
                                000030a00S0:00B000 缩0}0SFP
e00ft0!08V-T00T00000U0ks0F0am0001o00iP0 00akl000v};000j-0010Z0000H~0n00000000000067000|00_@-0x0y0S
n00`0nJ00 0;0zf0N7V·0'0s00k00%)000J0@0000I00s
2c/0o0000000000t
0!0η>0 0
    e0a0-J0m00 0000?0%
0S00Wy000000f0l0j0000p)]`pdbS0w0/08l000Di0\3tc000f00000(R+0o00[0
                                0rn0000_00"000fh0006RT0J00000=q0      0000S0
000000300000000000h[00/a0000000w0t
0000Z[0.>00Sk0;}000Kg000 F0F0000u0(0000I00[
                                A00B0l0dn'000}T0'3000U0 DT04000t+b70j000=[0aP0600
0100u 00[0'0+000000i00Ca000E0u0Cx0Q0Q000"+F00300+"0:(0jd0Dd00-g01wL0Z\0R8Hp0}0!0k0000=v|Ct07Uue0'0
```

Parece un socket lanzando basura, pero claro .. tiene que haber algo .. así que vamos a guardarlo en un fichero

```
root@kali:~/unaalmes# nc 34.247.69.86 1337 > prova.txt
```

Hago varias pruebas, de diferente duración para estar seguro y buscar patrones .. busco uam, UAM y cosas así dentro de los ficheros.

```
root@kali:~/unaalmes# strings prova2.txt | grep UAM
```

```
UAM:OWY5MTBhNjNiMGRINWMzNjM4YTA3MTg4MzFiN2JkODk0MGYxN2EyZjZjYTQ4MTE2
MDVIYmU0NGMwZjNkYjJiNmI2YzQzZjU1NmZhYjYwMWZ8a2V5OjFZRUFs
UAM:OWY5MTBhNjNiMGRINWMzNjM4YTA3MTg4MzFiN2JkODk0MGYxN2EyZjZjYTQ4MTE2
MDVIYmU0NGMwZjNkYjJiNmI2YzQzZjU1NmZhYjYwMWZ8a2V5OjFZRUFs
```

Como vemos, cada cierto tiempo el socket manda esta cadena.

```
root@kali:~/unaalmes# echo -n
"OWY5MTBhNjNiMGRINWMzNjM4YTA3MTg4MzFiN2JkODk0MGYxN2EyZjZjYTQ4MTE2MD
VIYmU0NGMwZjNkYjJiNmI2YzQzZjU1NmZhYjYwMWZ8a2V5OjFZRUFs" | base64 -d &&
echo
```

```
9f910a63b0de5c3638a0718831b7bd8940f17a2f6ca4811605ebe44c0f3db2b6b6c43f556fab601f
|key:1YEAR
```

Hasta aquí, había sido todo muy rápido y fácil ... pero se ha complicado. No encontraba el tipo de cifrado de la cadena ... Al final de mucho buscar, y volver a probar .. doy con una web que me da el resultado esperado:

<https://webnet77.net/cgi-bin/helpers/blowfish.pl>

The screenshot shows a web browser window with the title "BLOWFISH". The page contains instructions for using the tool and a form for encryption or decryption. The form has two radio buttons: "Encrypt" (selected) and "Decrypt". Below the radio buttons, there is a "Break at" dropdown menu set to "32" and a "Characters" label. The "Blowfish Key" field contains the text "1YEAR" and is labeled "MAX 56 Bytes". To the right of the key field, there is a red text label "padded with 3 bytes". Below the key field, there is a large blue box labeled "Blowfish Plain (or ASCII HEX if Encrypted)". Below this box, there are two empty text areas labeled "Blowfish Encrypted Text (Hexadecimal)".

FLAG: UAM{227218a71146ab9dc6ac28e5ec50a635}

Found : UAM_Ann1v3RS4R10

(hash = 227218a71146ab9dc6ac28e5ec50a635)

Felicidades a j0n3 por su first y también a BICACARO por el second ! Unos cracks todos !

**Muchas gracias a los organizadores felicidades por el AÑAZO y
esperemos que sean muchos muchos más !!!**

DarkEagle