

Silicon Valley - Episodio 3

UAM CTF 2018-11-15

El reto

<https://unaalmes.hispasec.com/challenges#EPISODIO%203>

Challenge

1 Solve



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}

TOP 3: 1. 2. 3.

Unlock Hint for 30 points

Unlock Hint for 60 points

Flag

Submit

Vamos a ver qué nos tienen preparado en este reto :)

Audio

Descargo y descomprimo el zip y veo que contiene un wav:

Avengers Infinity War Soundtrack - DEP Stan Lee.wav

Homenaje obligado al recientemente difunto Stan Lee y supongo que esconde alguna sorpresa pero... ¿dónde está la sorpresa?

Si lo escuchamos, es el tema aparentemente normal (ni pitiditos con morse ni Bella Ciao). Audacity no muestra nada relevante a la vista salvo, quizá un poco de ruido... ¿sospechoso?

A veces con un simple head o tail podemos ver cosas relevantes y en este caso, al hacer un tail, vemos algo al final:

tail Avengers Infinity War Soundtrack - DEP Stan Lee.wav



Parece que hay algo justo al final y tiene toda la pinta de ser un base64, así que decodifico:

```
> echo U29uaWRvUHJvZnVuZG87KQo= | base64 -d  
SonidoProfundo;)
```

¿SonidoProfundo? Quizá sea alguna técnica o algoritmo... no sé, no me suena... si fuera un algoritmo seguro que lo encuentro en inglés antes que en castellano, como casi todo :/

Busco "DeepSound crypto" en google y el primer resultado es:

jpinfo.net/deepsound/overview.aspx

jpinfo Home Game4You MyRepository DeepSound WebScamin Donate About

Overview
Download
Documentation

DeepSound overview

DeepSound is a steganography tool and audio converter that hides secret data into audio files. The application also enables you to extract secret files directly from audio files or audio CD tracks.

DeepSound might be used as copyright marking software for wave, flac, wma, ape, and audio CD. DeepSound also support encrypting secret files using AES-256(Advanced Encryption Standard) to improve data protection.

The application additionally contains an easy to use Audio Converter Module that can encode several audio formats (FLAC, MP3, WMA, WAV, APE) to others (FLAC, MP3, WAV, APE).

DeepSound reviews and video tutorials

MP3.com
Windows Software
EXCELLENT
DeepSound

DOWNLOAD NOW
instalki

System requirements:

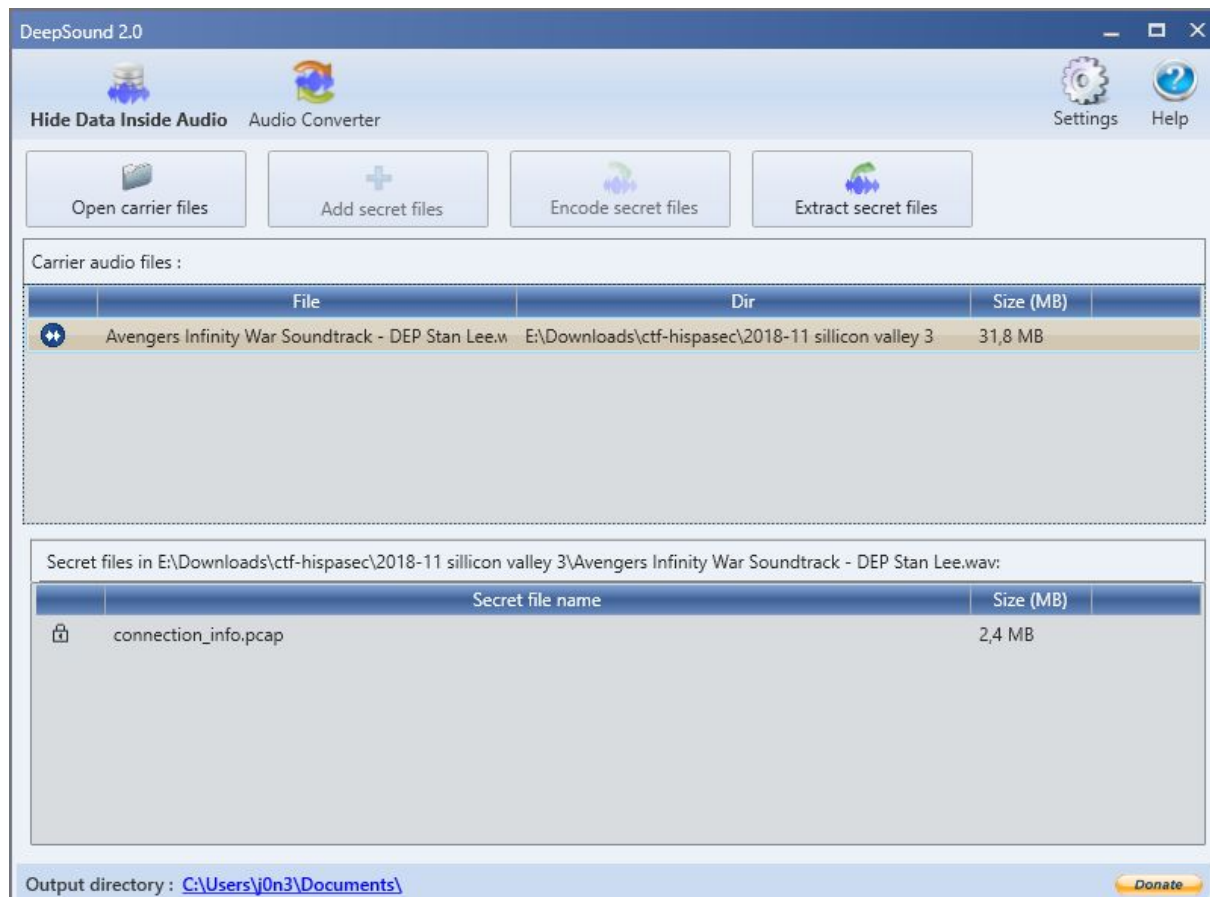
- Windows XP SP3/Vista/7/8/10
- Microsoft .NET Framework 4.0

Screenshots:

DeepSound 2.0

Me parece que sigo bien las pistas y creo que he acertado con el programa ;)

Descargar, instalar, ejecutar, abrir wav.

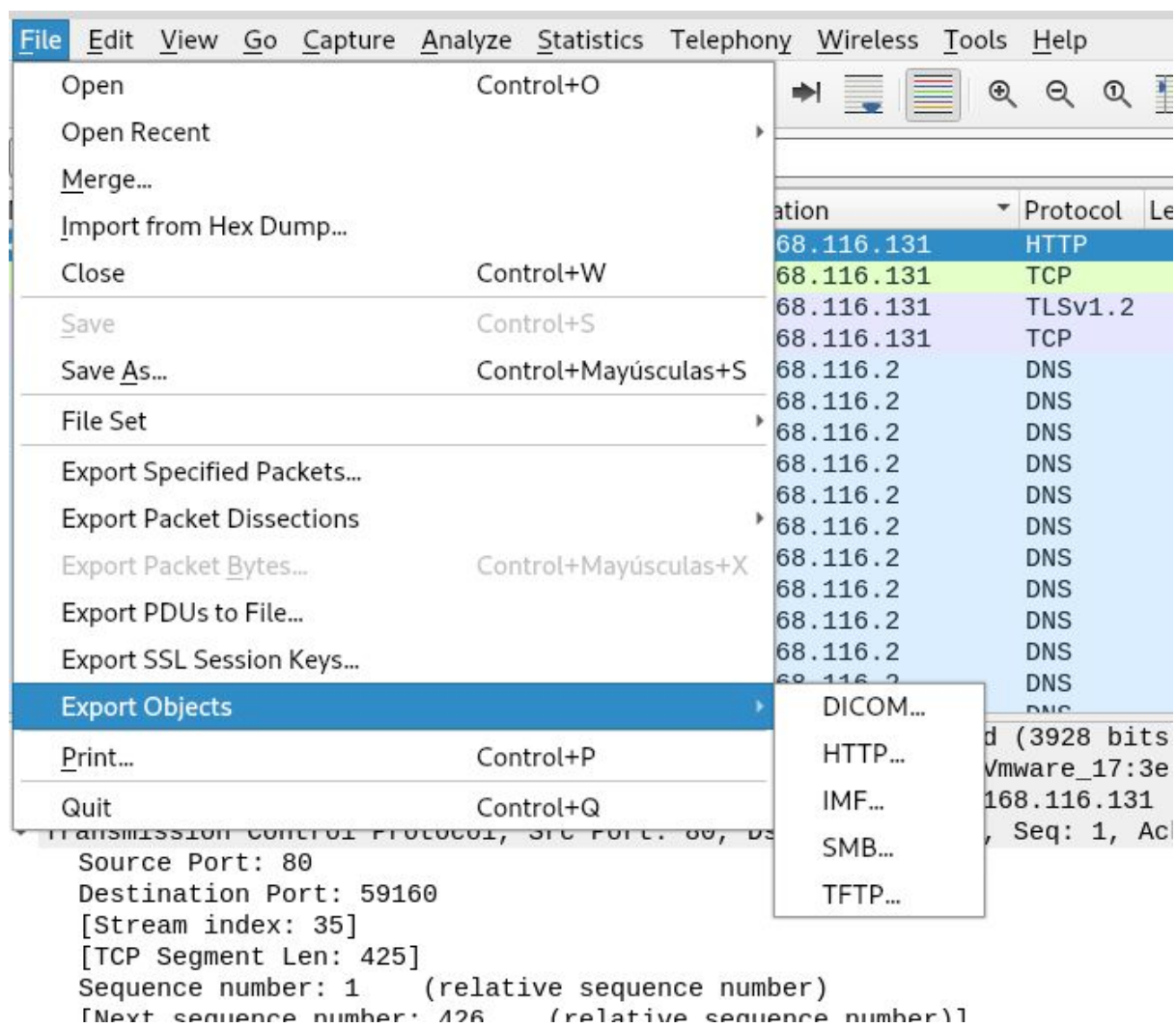


Parece que tenemos un archivo secreto.

connection_info.pcap

Aunque nunca he hecho análisis de pcap sí que recuerdo haberlo leído o visto por ahí (youtube, otros writeups de retos...). Sé que es un archivo que contiene una muestra de capturas de paquetes de tráfico de red y wireshark será nuestra herramienta. Exportaremos el pcap escondido en el wav y lo metemos en wireshark para analizarlo.

Tras un rato haciendo búsquedas de texto, mirando ip's y demás, descubro esto: si exportamos los objetos HTTP...



...vemos un archivo:

...y tras un rato capturando, buscando por 'uam' encuentro un paquete sospechoso.

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

src 34.247.69.86

Packet bytes Narrow & Wide Case sensitive String UAM

No.	Time	Source	Destination	Protocol	Length	Info
3962	36.931147867	10.0.2.15	34.247.69.86	TCP	54	60412 → 1337 [ACK] Seq=1 Ack=2358410 Win=65535 Len=0
3963	36.935168062	34.247.69.86	10.0.2.15	TCP	1514	1337 → 60412 [PSH, ACK] Seq=2358410 Ack=1 Win=65535 Len=1460
3964	36.935301352	10.0.2.15	34.247.69.86	TCP	54	60412 → 1337 [ACK] Seq=1 Ack=2359870 Win=65535 Len=0
3965	37.016209385	34.247.69.86	10.0.2.15	TCP	100	1337 → 60412 [PSH, ACK] Seq=2359870 Ack=1 Win=65535 Len=46
3966	37.016291087	10.0.2.15	34.247.69.86	TCP	54	60412 → 1337 [ACK] Seq=1 Ack=2359916 Win=65535 Len=0
3967	37.020345074	34.247.69.86	10.0.2.15	TCP	1514	1337 → 60412 [PSH, ACK] Seq=2359916 Ack=1 Win=65535 Len=1460
3968	37.020409583	10.0.2.15	34.247.69.86	TCP	54	60412 → 1337 [ACK] Seq=1 Ack=2361376 Win=65535 Len=0
3969	37.097403069	34.247.69.86	10.0.2.15	TCP	119	1337 → 60412 [PSH, ACK] Seq=2361376 Ack=1 Win=65535 Len=65
3970	37.097491589	10.0.2.15	34.247.69.86	TCP	54	60412 → 1337 [ACK] Seq=1 Ack=2361441 Win=65535 Len=0
3971	37.105795129	34.247.69.86	10.0.2.15	TCP	1514	1337 → 60412 [PSH, ACK] Seq=2361441 Ack=1 Win=65535 Len=1460
3972	37.105887442	10.0.2.15	34.247.69.86	TCP	54	60412 → 1337 [ACK] Seq=1 Ack=2362901 Win=65535 Len=0
3973	37.178515777	34.247.69.86	10.0.2.15	TCP	64	1337 → 60412 [PSH, ACK] Seq=2362901 Ack=1 Win=65535 Len=10
3974	37.178659119	10.0.2.15	34.247.69.86	TCP	54	60412 → 1337 [ACK] Seq=1 Ack=2362911 Win=65535 Len=0
3975	37.182385915	34.247.69.86	10.0.2.15	TCP	1514	1337 → 60412 [PSH, ACK] Seq=2362911 Ack=1 Win=65535 Len=1460
3976	37.182446898	10.0.2.15	34.247.69.86	TCP	54	60412 → 1337 [ACK] Seq=1 Ack=2364371 Win=65535 Len=0

Sequence number: 2362911 (relative sequence number)
 [Next sequence number: 2364371 (relative sequence number)]
 Acknowledgment number: 1 (relative ack number)
 0101 = Header Length: 20 bytes (5)
 Flags: 0x018 (PSH, ACK)
 Window size value: 65535
 [Calculated window size: 65535]
 [Window size scaling factor: -1 (unknown)]
 Checksum: 0x7a2a [unverified]
 [Checksum Status: Unverified]
 Urgent pointer: 0
 [SEQ/ACK analysis]
 [Timestamps]
 TCP payload (1460 bytes)
 Data (1460 bytes)
 Data: 2055414d3a4f5759354d5442684e6a4e694d47526c4e574d...

```

0030 ff ff 7a 2a 00 00 20 55 41 4d 3a 4f 57 59 35 4d  ..z*..UAM:OWY5M
0040 54 42 68 4e 6a 4e 69 4d 47 52 6c 4e 57 4d 7a 4e  TBhNjNiM GRlNWmZn
0050 6a 4d 34 59 54 41 33 4d 54 67 34 4d 7a 46 69 4e  jM4YTA3M Tg4MzFiN
0060 32 4a 6b 4f 44 6b 30 4d 47 59 78 4e 32 45 79 5a  2JkODk0M GYxN2EyZ
0070 6a 5a 6a 59 54 51 34 4d 54 45 32 4d 44 56 6c 59  jZjYTQ4M TE2MDVlY
0080 6d 55 30 4e 47 4d 77 5a 6a 4e 6b 59 6a 4a 69 4e  mU0NGMwZ jNkyJjiN
0090 6d 49 32 59 7a 51 7a 5a 6a 55 31 4e 6d 5a 68 59  mI2YzQzZ jU1NmZhY
00a0 6a 59 77 4d 57 5a 38 61 32 56 35 4f 6a 46 5a 52  jYwMWZ8a 2V50jFZR
00b0 55 46 53 20 37 19 6d 64 64 85 70 3d 57 8d 06 73  UFS 7 md d:p=W.s
00c0 4f 0f fc cc d7 e3 16 fb 41 bb 9a 5d 58 45 93 94  0.....A..]XE...
00d0 df 6b 5b c3 70 45 32 4b ea bb 42 69 a1 af 1a bb  .k[.pE2K..Bi....
00e0 66 a0 40 16 e6 53 e2 f3 bc af 45 1c ca d2 e6 d2  f@..S...E.....
00f0 a9 cb d8 d8 f9 76 13 cd d0 11 26 89 18 32 e8 30  ....v...&...2.0
0100 52 5a 60 6c 7e df 67 39 91 df 06 b8 45 d4 7f 63  RZ`l~g9...E..c
0110 80 76 6b 40 a8 f5 cc f2 fc 8f a5 66 33 a0 27 01  .vk@....-f3.'
0120 01 93 bb cd 68 75 53 33 2d 21 1d 79 5f a6 5a 7a  ...huS3 -!..y_.Zz
0130 30 67 f3 cd cf 8b 4f df 43 a4 db d4 fb a1 fc 88  0g....0..C.....
  
```

wireshark_eth0_2018116025504_DoTk42.pcapng

Extraigo la cadena completa y con dcode ([decodify](#)) pruebo si es algo que conozca y parece que es un base64



Al decodificarlo tenemos un hash y una key:

9f910a63b0de5c3638a0718831b7bd8940f17a2f6ca4811605ebe44c0f3db2b6b6c43f556fab601f|key:1YEAR

Probablemente sea un algoritmo de cifrado y 1YEAR sea la clave

blowfish

Tras muchas muchísimas vueltas, probando con decodificadores y encriptaciones varias, doy con ello. El método para encontrarlo fue buscar y probar todas y cada una de las herramientas que conozco y buscar unas cuantas más hasta dar con esta web de más abajo. Cybercheff no desencriptaba blowfish como esta:

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

Con casi cualquier herramienta de identificación de hashes dirá que coincide con RIPEMD320 pero por ahí no conseguí nada, así que seguí buscando herramientas hasta dar con esa. Si hubiera sabido que era blowfish me habría ahorrado varias horas de búsqueda de herramientas y prueba-error puesto que "blowfish decrypt" en google nos hubiera dado como primer resultado esta web... grrr

BLOWFISH

Blowfish is capable of strong encryption and can use key sizes up to 56 bytes (a 448 bit key). The key must be a multiple of 8 bytes (up to a maximum of 56). This example will automatically pad and unpad the key to size. Because Blowfish creates blocks of 8 byte encrypted output, the output is also padded and unpadded to multiples of 8 bytes.

❗ To Encrypt plain text Select "Encrypt" and paste the plain text in the "Blowfish Plain" box.

❗ To Decrypt, select "Decrypt", paste the ASCII-Hex encrypted text in in the "Blowfish Plain" box and make sure the password is the same as the one you used to Encrypt.

Encrypt/Decrypt ☐ Encrypt Break at Characters ☒ Decrypt

Blowfish Key
MAX 56 Bytes

1YEAR

9f910a63b0de5c3638a0718831b7bd8940f17a2f6ca4811605ebe44c0f3db2b6b6c43f556fab601f

Blowfish
Plain (or ASCII
HEX if
Encrypted)

Blowfish
Encrypted Text
(Hexadecimal)

BLOWFISH

Blowfish is capable of strong encryption and can use key sizes up to 56 bytes (a 448 bit key). The key must be a multiple of 8 bytes (up to a maximum of 56). This example will automatically pad and unpad the key to size. Because Blowfish creates blocks of 8 byte encrypted output, the output is also padded and unpadded to multiples of 8 bytes.

i To **Encrypt** plain text Select "Encrypt" and paste the plain text in the "Blowfish Plain" box.
 i To **Decrypt**, select "Decrypt", paste the ASCII-Hex **encrypted** text in the "Blowfish Plain" box and make sure the password is the same as the one you used to Encrypt.

Encrypt/Decrypt ☒ Encrypt Break at Characters ☐ Decrypt

Blowfish Key
 MAX 56 Bytes

1YEAR padded with 3 bytes

UAM{227218a71146ab9dc6ac28e5ec50a635}

Blowfish Plain (or ASCII HEX if Encrypted)

Nothing to do

Nothing to do

Blowfish Encrypted Text (Hexadecimal)

Flag

UAM{227218a71146ab9dc6ac28e5ec50a635}

Lo meto en la plataforma y... ¡First blood! ¡Yujuuu!

Challenge	1 Solves	X
Name	Date	
j0n3	a few seconds ago	

Conclusión

Ha sido un ctf muy variado, con esteganografía, cifrados, análisis y captura de tráfico de red de servidores indiscretos... y un crypto traicionero que me ha costado horas por confiar en mis herramientas habituales. He aprendido a analizar un pcap con wireshark, a no confiar ciegamente en las herramientas y a volver a probar lo que ya estaba probado. Parece que a veces caminar en círculos no es tan mala idea ;)

Gracias a @mrb0b0t y @devploit por este ctf tan currado :D

José Ángel Sánchez

[@_j0n3](#)