UAM Reto Silicon Valley: Episodio 2 - Hispasec

Descripción

Nombre: UAM- Silicon Valley - Episodio 2 - (Related https://www.filmaffinity.com/es/film279751.html)
Fecha de liberación: 15 de septiembre de 2018
Autor: 1v4n https://unaalmes.hispasec.com/team/40
Puntuación: 300

Dinesh ha perdido la clave VERDADERA que usaba para abrir su zip secreto pero gracias a DIOS tiene un archivo .raw donde puede recuperarla y necesita que le echemos una mano.

A Dinesh le encantan los mensajes con doble sentido, debéis tenerlo en cuenta...

Archivo .raw (escoged el que mejor os venga):
https://www.mediafire.com/file/piv4t8514bp5dpg/pied_piper_bak.zip/file

https://mega.nz/#!iAUDnKwA!Y2g23qnZ9rwZvzZA3Bg8cbENe_ZtASOi1NFgrgfL8sg

Info: Las pistas os servirán a partir de que tengáis la contraseña del zip adjunto (Secretos_Dinesh.zip). Recordad que flag.txt tiene dos cifrados (leed bien README).
https://unaalmes.hispasec.com/files/79df65e53ab8565419d8105b6363d03f/Secretos_Dinesh.zip

Objetivo

Formato de la flag: UAM{md5}

Herramientas utilizadas

Versión 69.0.3497.100 (Build oficial) (64 bits) https://www.google.com/chrome/
megatools 1.10.2 - command line tools for Mega.nz https://github.com/megous/megatools
file-5.34
UnZip 6.00 ftp://ftp.info-zip.org/pub/infozip/
TestDisk 7.0, Data Recovery Utility, April 2015
curl 7.61.0
Volatility Foundation Volatility Framework 2.6
https://www.dcode.fr/bacon-cipher
https://gchq.github.io/CyberChef

Resumen:

Comenzamos por visitar el reto y descargamos el archivo adjunto *con el Disco duro de Gilfoyle* utilizando la tool *megadl 'https://mega.nz/#!iAUDnKwA!Y2g23qnZ9rwZvzZA3Bg8cbENe_ZtASOi1NFgrgfL8sg'*

```
root@kali:~/Desktop/uam/SiliconValley# root@kali:~/Desktop/uam/SiliconValley#
megadl 'https://mega.nz/#!iAUDnKwA!Y2g23qnZ9rwZvzZA3Bg8cbENe_ZtASOi1NFgrgfL8sg'
Downloaded pied_piper_bak.zip
```

Procesado de archivo dump

Descargamos el artefacto *pied_pier_bak.zip* de 660,2 MB que descomprimido con *unzip* obtenemos *pied_piper_bak.raw* de 1,1 GB y le pasamos un *testdisk*

```
root@kali:~/Desktop/uam/SiliconValley# unzip pied_piper_bak.zip
Archive:  pied_piper_bak.zip
  inflating: pied_piper_bak.raw
root@kali:~/Desktop/uam/SiliconValley# file pied_piper_bak.raw
pied_piper_bak.raw: data
root@kali:~/Desktop/uam/SiliconValley# testdisk /list pied_piper_bak.raw
TestDisk 7.0, Data Recovery Utility, April 2015
Christophe GRENIER <grenier@cgsecurity.org>
http://www.cgsecurity.org
Please wait...
Disk pied_piper_bak.raw - 1098 MB / 1047 MiB - CHS 134 255 63
Sector size:512


Disk pied_piper_bak.raw - 1098 MB / 1047 MiB - CHS 134 255 63
     Partition                  Start        End    Size in sectors
```

Recordamos la referencia en la primera misión de los retos de UAM y en el articulo del Team bi0s https://amritabi0s.wordpress.com/2017/09/24/sec-t-ctf-g1bs0n-writeup/ .Por lo tanto pasamos a utilizar Volatility:

```
root@kali:~/Desktop/uam/SiliconValley# volatility imageinfo -f
pied_piper_bak.raw
Volatility Foundation Volatility Framework 2.6
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : Win7SP1x64, Win7SP0x64, Win2008R2SP0x64,
Win2008R2SP1x64_23418, Win2008R2SP1x64, Win7SP1x64_23418
                     AS Layer1 : WindowsAMD64PagedMemory (Kernel AS)
                     AS Layer2 : FileAddressSpace
(/root/Desktop/uam/SiliconValley/pied_piper_bak.raw)
                      PAE type : No PAE
                           DTB : 0x187000L
                          KDBG : 0xf80002a520a0L
          Number of Processors : 1
     Image Type (Service Pack) : 1
                KPCR for CPU 0 : 0xfffff80002a53d00L
             KUSER_SHARED_DATA : 0xfffff78000000000L
           Image date and time : 2018-10-15 10:48:27 UTC+0000
     Image local date and time : 2018-10-15 12:48:27 +0200
```

Seleccionamos el perfil 'Win2008R2SP0x64' y pasamos a analizar los procesos que se estaban ejecutando en el momento que fue tomado el "dump":

```
root@kali:~/Desktop/uam/SiliconValley# volatility pslist
--profile=Win2008R2SP0x64 -f pied_piper_bak.raw
Volatility Foundation Volatility Framework 2.6
Offset(V)          Name                    PID   PPID   Thds     Hnds   Sess
Wow64 Start                 Exit
------------------ -------------------- ------ ------ ------ -------- ------
------ ------------------------------ ------------------------------
0xfffffa8000ce69e0 System                    4      0     88      529 ------
0 2018-10-15 10:16:23 UTC+0000
0xfffffa8001c85130 smss.exe                268      4      2       29 ------
0 2018-10-15 10:16:23 UTC+0000
0xfffffa8002737b30 csrss.exe               352    344      9      394      0
0 2018-10-15 10:16:28 UTC+0000
0xfffffa8000cea670 wininit.exe             400    344      3       74      0
0 2018-10-15 10:16:28 UTC+0000
0xfffffa8000ced100 csrss.exe               412    392      7      223      1
0 2018-10-15 10:16:28 UTC+0000
0xfffffa8002749700 winlogon.exe            452    392      5      113      1
0 2018-10-15 10:16:28 UTC+0000
0xfffffa800276eb30 services.exe            496    400     10      203      0
0 2018-10-15 10:16:28 UTC+0000
0xfffffa800274bb30 lsass.exe               516    400      8      724      0
0 2018-10-15 10:16:30 UTC+0000
0xfffffa800277ab30 lsm.exe                 524    400     10      144      0
```

```
0 2018-10-15 10:16:30 UTC+0000
0xfffffa800277cb30 svchost.exe             628    496     11      351      0
0 2018-10-15 10:16:31 UTC+0000
0xfffffa80027f6060 VBoxService.ex          688    496     12      118      0
0 2018-10-15 10:16:31 UTC+0000
0xfffffa8002815b30 svchost.exe             740    496      9      260      0
0 2018-10-15 10:16:31 UTC+0000
0xfffffa800287a5b0 svchost.exe             868    496     24      576      0
0 2018-10-15 10:16:31 UTC+0000
0xfffffa800287db30 svchost.exe             928    496     26      526      0
0 2018-10-15 10:16:31 UTC+0000
0xfffffa800289ab30 svchost.exe             968    496     30      888      0
0 2018-10-15 10:16:31 UTC+0000
0xfffffa80028dbb30 svchost.exe             312    496     16      435      0
0 2018-10-15 10:16:32 UTC+0000
0xfffffa800291c360 svchost.exe            1088    496     16      364      0
0 2018-10-15 10:16:32 UTC+0000
0xfffffa800298d390 spoolsv.exe            1188    496     13      280      0
0 2018-10-15 10:16:32 UTC+0000
0xfffffa80029cc2e0 svchost.exe            1228    496     19      307      0
0 2018-10-15 10:16:33 UTC+0000
0xfffffa8002a0db30 svchost.exe            1356    496     21      296      0
0 2018-10-15 10:16:33 UTC+0000
0xfffffa8002a4ab30 GRRservice.exe         1400    496      3       41      0
0 2018-10-15 10:16:33 UTC+0000
0xfffffa8002a51b30 GRR.exe                1432   1400      6      268      0
0 2018-10-15 10:16:33 UTC+0000
0xfffffa8002c4c060 taskhost.exe           1060    496      9      205      1
0 2018-10-15 10:18:47 UTC+0000
0xfffffa8002891a10 dwm.exe                1336    928      3       70      1
0 2018-10-15 10:18:47 UTC+0000
0xfffffa8000e2cb30 explorer.exe           1464   1284     31      886      1
0 2018-10-15 10:18:47 UTC+0000
0xfffffa8000ee7b30 VBoxTray.exe           1572   1464     12      148      1
0 2018-10-15 10:18:48 UTC+0000
0xfffffa8002882760 jusched.exe            1820   1552      2       67      1
1 2018-10-15 10:18:50 UTC+0000
0xfffffa800285ab30 SearchIndexer.         1928    496     11      563      0
0 2018-10-15 10:18:57 UTC+0000
0xfffffa8000e95b30 sppsvc.exe              836    496      4      141      0
0 2018-10-15 10:19:27 UTC+0000
0xfffffa8000dd57a0 svchost.exe             892    496     13      317      0
0 2018-10-15 10:19:27 UTC+0000
0xfffffa8000e73b30 wmpnetwk.exe           2056    496     13      438      0
0 2018-10-15 10:19:28 UTC+0000
0xfffffa8000ddb2a0 svchost.exe            3048    496      9      346      0
0 2018-10-15 10:22:07 UTC+0000
0xfffffa80012d9b30 DB Browser for         1836   1464      9      345      1
0 2018-10-15 10:26:16 UTC+0000
0xfffffa8001355060 notepad.exe            2616   1464      1       62      1
```

```
0 2018-10-15 10:29:08 UTC+0000
0xfffffa8002b7a1b0 cmd.exe               2312   1464      1       21      1
0 2018-10-15 10:32:00 UTC+0000
0xfffffa8001273b30 conhost.exe           2200    412      2       51      1
0 2018-10-15 10:32:00 UTC+0000
0xfffffa80013286e0 notepad.exe           1520   1464      1       62      1
0 2018-10-15 10:39:42 UTC+0000
0xfffffa8001230060 audiodg.exe           2880    868      6      126      0
0 2018-10-15 10:48:25 UTC+0000
0xfffffa800103db30 DumpIt.exe            2968   1464      1       25      1
1 2018-10-15 10:48:26 UTC+0000
0xfffffa8000fd7570 conhost.exe            824    412      2       51      1
0 2018-10-15 10:48:26 UTC+0000
```

Descubrimos revisando los procesos y comprobando que hay dos procesos que nos llaman la atención como *"notepad.exe"* y *"DB Browser for"*. Usando el plugin *filescan* nos arroja archivos ".txt" lo siguiente con un *grep*

```
root@kali:~/Desktop/uam/SiliconValley# volatility filescan
--profile=Win2008R2SP0x64 -f pied_piper_bak.raw | grep .txt
Volatility Foundation Volatility Framework 2.6
0x000000004146c660     16      0 R--rwd
\Device\HarddiskVolume2\Users\Richard\Desktop\piper.txt
0x0000000041515630      1      1 -W-rw-
\Device\HarddiskVolume2\Users\Richard\AppData\Local\Temp\FXSAPIDebugLogFile.txt
```

Pasamos a intentar extraer el archivo piper.txt pero nos encontramos problemas

```
root@kali:~/Desktop/uam/SiliconValley# volatility --profile=Win2008R2SP0x64 -f
pied_piper_bak.raw dumpfiles -Q 0x000000004146c660 --name -D
/root/Desktop/uam/SiliconValley/
Volatility Foundation Volatility Framework 2.6
DataSectionObject 0x4146c660   None
\Device\HarddiskVolume2\Users\Richard\Desktop\piper.txt
```

No obtenemos nada pero el nombre del archivo *"piper"* puede orientarnos a como pista para dar con el archivo que tiene las credenciales. Pasamos a usar de nuevo el plugin *filescan* con un grep *"piper"*

```
root@kali:~/Desktop/uam/SiliconValley# volatility filescan
--profile=Win2008R2SP0x64 -f pied_piper_bak.raw | grep "piper"
Volatility Foundation Volatility Framework 2.6
0x0000000040501860     16      0 R--rw-
\Device\HarddiskVolume2\Users\Richard\Desktop\piperdb.db
0x000000004054bf20      4      0 RW---- \Device\HarddiskVolume2\Program Files
(x86)\DumpIt\piedpiper_bak.zip15-104032.zip
0x000000004123a8d0      1      1 RW-rw-
\Device\HarddiskVolume2\Users\Richard\Desktop\piperdb.db
0x000000004146c660     16      0 R--rwd
\Device\HarddiskVolume2\Users\Richard\Desktop\piper.txt
```

Pasamos a intentar extraer el archivo piperdb.db y con éxito obtenemos 2 archivos con extensión .db que analizamos

```
root@kali:~/Desktop/uam/SiliconValley# volatility --profile=Win2008R2SP0x64 -f
pied_piper_bak.raw dumpfiles -Q 0x0000000040501860 --name -D
/root/Desktop/uam/SiliconValley/
Volatility Foundation Volatility Framework 2.6
DataSectionObject 0x40501860   None
\Device\HarddiskVolume2\Users\Richard\Desktop\piperdb.db
SharedCacheMap 0x40501860   None
\Device\HarddiskVolume2\Users\Richard\Desktop\piperdb.db
```

```
root@kali:~/Desktop/uam/SiliconValley# strings
file.None.0xfffffa80013806d0.piperdb.db.vacb
SQLite format 3
atableCOMMUNICATIONSCOMMUNICATIONS
CREATE TABLE `COMMUNICATIONS` (
       `idmsg`      INTEGER,
       `msg`  TEXT,
       `rcv`  INTEGER,
       `user` TEXT,
       `sum`  TEXT
%tableFLAGFLAG
CREATE TABLE `FLAG` (
       `id_flag`    INTEGER,
       `char_flag`  TEXT,
       `falso`      TEXT
_tableUSERSUSERS
CREATE TABLE `USERS` (
       `id`   INTEGER UNIQUE,
       `user` TEXT,
       `pass` TEXT,
       `age`  INTEGER,
       `md5`  INTEGER
indexsqlite_autoindex_USERS_1USERS
true_godBAAABAABAAABBAAAAAAABBAABABBBAAAAAAAAAABAAAAAAAABAABBBAABBABABAAAAAAAAAB
BAB3L_R3m1t0_m3j0R_Q_L4_fl4ut4,
ghost71144850f4fb4cc55fc0ee6935badddfL
jaredecd5c54d0956b37daff84de64e06326fecd5c54d0956b37daff84de64e06326fM
erliche3353512022242b52c702b4b38951356e3353512022242b52c702b4b389513560
gilfoyle327a6c4304ad5938eaf0efb6cc3e53dc327a6c4304ad5938eaf0efb6cc3e53dcN
richard97a53ee9f45adfe53c762a72f83f6f4397a53ee9f45adfe53c762a72f83f6f43L
Madmin21232f297a57a5a743894a0e4a801fc3
21232f297a57a5a743894a0e4a801fc3
root@kali:~/Desktop/uam/SiliconValley# strings
file.None.0xfffffa800273c2d0.piperdb.db.dat
SQLite format 3
atableCOMMUNICATIONSCOMMUNICATIONS
CREATE TABLE `COMMUNICATIONS` (
       `idmsg`      INTEGER,
```

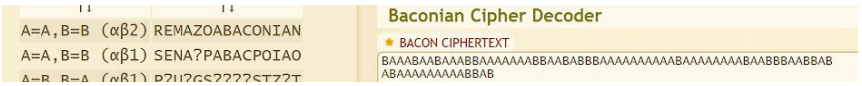
```
	`msg`	TEXT,
	`rcv`	INTEGER,
	`user`	TEXT,
	`sum`	TEXT
%tableFLAGFLAG
CREATE TABLE `FLAG` (
	`id_flag`	INTEGER,
	`char_flag`	TEXT,
	`falso`	TEXT
_tableUSERSUSERS
CREATE TABLE `USERS` (
	`id`	INTEGER UNIQUE,
	`user`	TEXT,
	`pass`	TEXT,
	`age`	INTEGER,
	`md5`	INTEGER
indexsqlite_autoindex_USERS_1USERS
true_godBAAABAABAAABBAAAAAAABBAABABBBAAAAAAAAAABAAAAAAAABAABBBAABBABABAAAAAAAAAB
BAB3L_R3m1t0_m3j0R_Q_L4_fl4ut4,
ghost71144850f4fb4cc55fc0ee6935badddfL
jaredecd5c54d0956b37daff84de64e06326fecd5c54d0956b37daff84de64e06326fM
erliche3353512022242b52c702b4b38951356e3353512022242b52c702b4b389513560
gilfoyle327a6c4304ad5938eaf0efb6cc3e53dc327a6c4304ad5938eaf0efb6cc3e53dcN
richard97a53ee9f45adfe53c762a72f83f6f4397a53ee9f45adfe53c762a72f83f6f43L
Madmin21232f297a57a5a743894a0e4a801fc3
21232f297a57a5a743894a0e4a801fc3
root@kali:~/Desktop/uam/SiliconValley#
```

Obtención de credenciales para descomprimir

Vemos que obtenidos los strings de los dos archivos obtenidos detectamos una cadena que nos llama la atención:

```
true_godBAAABAABAAABBAAAAAAABBAABABBBAAAAAAAAAABAAAAAAAABAABBBAABBABABAAAAAAAAAB
BAB3L_R3m1t0_m3j0R_Q_L4_fl4ut4
```

Pasamos a decodificar a través de la herramienta online https://www.dcode.fr/bacon-cipher que nos arroja una posible clave **REMAZOABACONIAN**



Descargamos el archivo *Secretos_Dinesh.zip* de 654 bytes que pasamos a descomprimir con ayuda de la *wget* https://unaalmes.hispasec.com/files/79df65e53ab8565419d8105b6363d03f/Secretos_Dinesh.zip

```
root@kali:~/Desktop/uam/SiliconValley# unzip Secretos_Dinesh.zip
Archive:  Secretos_Dinesh.zip
[Secretos_Dinesh.zip] flag.txt password:
  inflating: flag.txt
  inflating: README
```

Decodificación

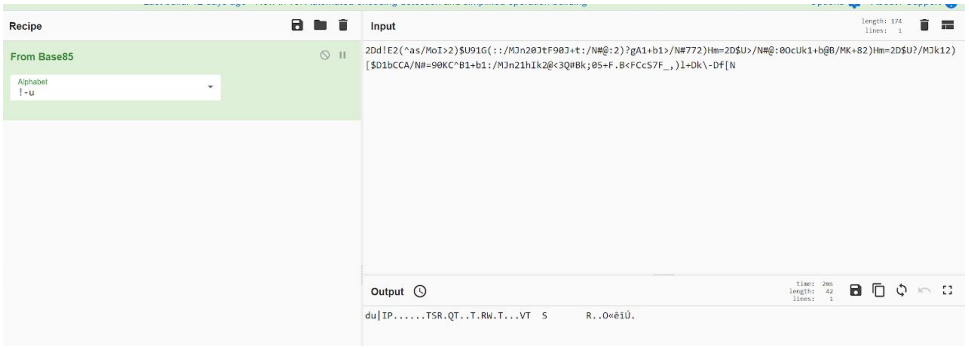
Obtenemos 2 archivos uno README, nos aporta claves para decodificar la cadena *"2Dd!E2(^as/MoI>2)$U91G(::/MJn20JtF90J+t:/N#@:2)?gA1+b1>/N#772)Hm=2D$U>/N#@:0OcUk1+b@B/MK+82)Hm=2D$U?/MJk12)[$D1bCCA/N#=90KC^B1+b1:/MJn21hIk2@<3Q#Bk;05+F.B<FCcS7F_,)l+Dk\-Df[N"*

```
root@kali:~/Desktop/uam/SiliconValley# cat README
1. "We are the DATE" https://www.youtube.com/watch?v=tYIYRRLj-n4

2. La clave final de todo está en el corazón de Telegram, en sus comienzos…

root@kali:~/Desktop/uam/SiliconValley# cat flag.txt
2Dd!E2(^as/MoI>2)$U91G(::/MJn20JtF90J+t:/N#@:2)?gA1+b1>/N#772)Hm=2D$U>/N#@:0OcUk
1+b@B/MK+82)Hm=2D$U?/MJk12)[$D1bCCA/N#=90KC^B1+b1:/MJn21hIk2@<3Q#Bk;05+F.B<FCcS7
F_,)l+Dk\-Df[N
```

Primero visitamos el video del servicio YouTube en la URL https://www.youtube.com/watch?v=tYIYRRLj-n4 con el título "USA for Africa - We Are The World - 1985" y que con orientación de los admins el "85" es la parte importante en el DATE >_ Decodificamos obteniendo de Base85 *"64-75-7c-49-50-03-03-01-05-00-06-54-53-52-08-51-54-06-04-54-0b-52-57-07-54-06-05-00-56-54-09-53-09-52-04-01-4f Vas bien, ya te queda menos."*



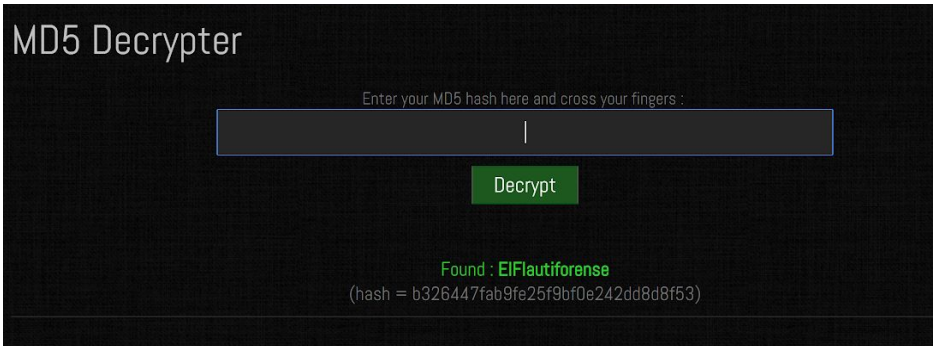
Vamos por buen camino .Y observamos que tenemos un Hexadecimal que pasaremos también a decodificar

A partir de aquí volvemos a orientarnos de nuevo conociendo que la cadena "du|IPTSRQTT RWTVT S RO" está encriptada con XOR, pero con una clave que el segundo consejo del README nos ayudará "La clave final de todo está en el corazón de Telegram, en sus comienzos…". Dándole vueltas a nuestra clave es el DATE del comienzo del grupo de la UAM https://web.telegram.org/#/im?p=%40unaalmes con el formato DDMMAAAA que fue el 14122017

Obtención de la Flag

Pasamos el desencriptar el XOR con la clave 14122017:

Y la solución es ***UAM{3b92d18aa7a6176dd37d372bc2f1eb71}***



Autor: MXY0bg== a.k.a. 1v4n

Twitter: https://twitter.com/Hackers4f // https://twitter.com/1r0Dm48O