root@kali:~/Security/IMF# nmap -sT -Pn -sV -T5 192.168.0.9

Starting Nmap 7.01 (https://nmap.org) at 2016-12-18 12:22 CET
Nmap scan report for 192.168.0.9
Host is up (0.00053s latency).
Not shown: 997 filtered ports
PORT   STATE SERVICE VERSION
22/tcp closed ssh
80/tcp open   http   Apache httpd 2.4.18 ((Ubuntu))
443/tcp open   ssl/http Apache httpd 2.4.18 ((Ubuntu))

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 16.43 seconds

Siguiendo la pista del flag1, entramos en la página de Home y mirando el código para ver si encontramos el Hexadecimal, vemos que:

<!--[if IE 8]> <html lang="en" class="ie8"> <![endif]-->
<!--[if IE 9]> <html lang="en" class="ie9"> <![endif]-->
<!--[If IE4]><script src="/oldIE/html5.js"></script><![Make sure to remove this before going to PROD]-->
<!--[if !IE]><!-→

Ergo, miremos el javascript al que se nos hace referencia, la primera línea pone.

/* 666c61677b376330313332303730613065663731643534323636336539646331663564656 57d */

Convirtiendo el hex a string con la página: http://string-functions.com/hex-string.aspx
Obtenemos una flag: flag{7c0132070a0ef71d542663e9dc1f5dee}

Intentemos descrifrar este hash MD5 con: https://hashkiller.co.uk/md5-decrypter.aspx
Que nos da la palabra: nmap

Así pues, pasemos el nmap de nuevo pero para todos los puertos esta vez:

root@kali:~/Security/IMF# nmap -sT -Pn -sV -p 1-65535 -T5 192.168.0.9

Starting Nmap 7.01 (https://nmap.org) at 2016-12-18 13:15 CET
Stats: 0:00:01 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 0.01% done
Nmap scan report for 192.168.0.9
Host is up (0.00047s latency).
Not shown: 65531 filtered ports
PORT   STATE SERVICE VERSION
22/tcp   closed ssh
80/tcp   open   http   Apache httpd 2.4.18 ((Ubuntu))
443/tcp   open   ssl/http Apache httpd 2.4.18 ((Ubuntu))
22222/tcp open   ssh   OpenSSH 7.2p2 Ubuntu 4ubuntu2.1 (Ubuntu Linux; protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

```
root@kali:~/Security/IMF# ssh -p 22222 192.168.0.9
The authenticity of host '[192.168.0.9]:22222 ([192.168.0.9]:22222)' can't be established.
ECDSA key fingerprint is
SHA256:DeCMZ74o5wesBHFLyaVY7UTCA7mW+bx6WroHm6AgMqU.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[192.168.0.9]:22222' (ECDSA) to the list of known hosts.
##################################################################
#                     WARNING                    #
#                  FBI - Authorized access only!              #
# Disconnect IMMEDIATELY if you are not an authorized user!!! #
#        All actions Will be monitored and recorded         #
#         Flag{53c82eba31f6d416f331de9162ebe997}                    #
##################################################################
root@192.168.0.9's password:
```

https://www.md5lab.com/md5/53c82eba31f6d416f331de9162ebe997

53c82eba31f6d416f331de9162ebe997 = encrypt

Tras no entender a qué se refiere con esto, intentando meter la contraseña “encrypt”, encrypt hasheado con diferentes algoritmos y demás…

Accedemos por https a la página del reto y en ver más información del certificado, aparece el flag:

f82366a9ddc064585d54e3f78bde3221 = personnel

De hacer el directory browsing/forcing con dirb, gobuster y ZAP, recordaba este directorio. Al intentar acceder:

http://192.168.0.9/personnel/

ACCESS DENIED!!! You Do Not Appear To Be Coming From An FBI Workstation. Preparing Interrogation Room 1. Car Batteries Charging.…

Lo que nos dice el flag es:

Flag #4 A Good Agent is Hard to Find.

Está bastante claro que se está refiriendo al user-agent adecuado para que vea que proviene del FBI. Después de probar con unos cuántos (FBI, FBI workstation, Federal Bureu Investigation…), recordamos que el javascript de dónde hemos sacado antes la pista de nmap, necesitaba el viejo internet explorer 4.

Miremos pues este javascript otra vez. Esta vez, para verlo con un formato adecuado, podemos embellecerlo aquí: http://jsbeautifier.org/

Es un archivo enorme pero examinándolo un poco, vemos el siguiente comentario:

/* maindev - 6/7/02 Adding temporary support for IE4 FBI Workstations */
/* newmaindev - 5/22/16 Last maindev was and idoit and IE4 is still Gold image -@Support doug.perterson@fbi.gov */

Así pues, copiamos la cadena que identifica al User-Agent IE4 de aquí:
http://www.useragentstring.com/index.php?id=3110

Y con Burp, interceptando la petición, sustituimos el user-agent de nuestro navegador por éste. De esta forma ya tendremos acceso a la página.

Aquí aparece la nueva flag:

14e10d570047667f904261e6d08f520f MD5 : evidence

Y como nos dicen que Clue = new+flag, "newevidence"

En primera instancia, pienso que newevidence puede ser la contraseña para la conexión ssh de antes, pero no resulta serlo. Podría tratarse de un directorio, tal que http://192.168.0.9/newevidence

Al intentar entrar en ese directorio, vuelve a salir el aviso:

ACCESS DENIED!!! You Do Not Appear To Be Coming From An FBI Workstation. Preparing Interrogation Room 1. Car Batteries Charging.…

Cambiando otra vez el user-agent por el de internet explorer 4, nos sale una pantall de login. De la dirección de email que hemos encontrado antes en el javascript, podemos inferir que los nombres de usuario son del tipo nombre.apellido@fbi.gov y de la pista del flag 4, entendemos que la contraseña es algo personal y fácilmente adivinable.
Necesitaremos buscar en internet cuál era el nombre completo del agente en la película Atrápame si puedes. Es fácil ver que su nombre completo era: Carl Hanratty, así pues, tenemos que será carl.hanratty@fbi.gov o carl.hanratty sin el dominio.

Siguiendo la pista del flag 5, buscamos en internet diálogos de la película para localizar si el agente Carl habla de algo personal y que pueda parecerse a un password en la película.
Después de revisar diálogos en wikiquote y en imdb, la primera posibilidad que me viene a la cabeza (junto con otras tantas), es el nombre de su hija, Grace. Tras probar varias combinaciones, la ganadora resulta ser:

carl.hanratty/Grace

Una vez logrado el acceso, si hacemos clic en el enlace de la evidencia:
http://192.168.0.9/newevidence/Evidence.txt

117c240d49f54096413dd64280399ea9 MD5 : panam

Subiendo la imagen que encontramos en el link de http://192.168.0.9/newevidence/image.jpg al buscador de imágenes de Google, descubrimos que se trata de una imagen de Montrichard, en el valle del Loira (Francia)

Además, intentando extraer información de la imagen en mi máquina local:

root@kali:~/Security/Skydog# strings image.jpg | less (Esto apenas da información útil)
root@kali:~/Security/Skydog# exiftool image.jpg
ExifTool Version Number : 10.23
File Name : image.jpg
Directory : .
File Size : 4.1 MB
File Modification Date/Time : 2016:12:19 11:48:16+01:00
File Access Date/Time : 2016:12:19 11:50:58+01:00
File Inode Change Date/Time : 2016:12:19 11:48:16+01:00
File Permissions : rw-r--r--
File Type : JPEG
File Type Extension : jpg
MIME Type : image/jpeg
JFIF Version : 1.01
Resolution Unit : None
X Resolution : 1
Y Resolution : 1
Image Width : 3456
Image Height : 2304
Encoding Process : Baseline DCT, Huffman coding
Bits Per Sample : 8
Color Components : 3
Y Cb Cr Sub Sampling : YCbCr4:2:2 (2 1)
Image Size : 3456x2304
Megapixels : 8.0
root@kali:~/Security/Skydog# binwalk -B image.jpg

DECIMAL HEXADECIMAL DESCRIPTION
---
0 0x0 JPEG image data, JFIF standard 1.01
2214320 0x21C9B0 MySQL MISAM compressed data file Version 10

Me doy cuenta de que siguiendo por este camino la cosa se complica bastante, cuando no es realmente imposible avanzar, así que hay que pensar en otro vector de ataque.

El recibo en pdf en principio tampoco nos ofrece ninguna pista aprovechable.

La pista del flag anterior decía “panam” que no “Panama”. Buscando en google que puede significar esto, la primera entrada nos habla de la compañía aérea Pan Am:
https://es.wikipedia.org/wiki/Pan_Am

Kali no tiene aplicaciones específicamente dedicadas a la esteganografía, así que instalamos “steghide” para estos menesteres.
Mirando su manual de ayuda, vemos que para extraer el contenido de la imagen que hemos bajado, el comando a utilizar es:
root@kali:~/Security/Skydog# steghide --extract -sf image.jpg
Anotar salvoconducto:
anot los datos extrados e/"flag.txt".

El salvoconducto, en este caso hace referencia a una contraseña, es “panam”, la primera palabra que

hemos probado.
Así pues, recapitulando, tenemos que el flag es: ILoveFrance
La pista es: iheartbrenda
Y el flag 7 inicial es: Flag #7 Frank Was Caught on Camera Cashing Checks and Yelling - I'm The Fastest Man Alive!

Cómo no hay por donde cogerlo, busquemos en google. Parece hacer referencia a una frase de la serie Flash Gordon: FLASH - My Name is Barry Allen, and I'm the fastest man alive
Así las cosas, puede ser que Barry Allen sea un nuevo nombre de usuario. Probamos a loguearnos con barry.allen y como contraseña IloveFrance/iheartbrenda, pero ninguna funciona.
Si probamos la combinación barryallen/iheartbrenda, entonces conseguimos el acceso deseado:

```
root@kali:~# ssh -p 22222 barryallen@192.168.0.9
##################################################################
#                     WARNING                    #
#                  FBI - Authorized access only!              #
# Disconnect IMMEDIATELY if you are not an authorized user!!! #
#        All actions Will be monitored and recorded         #
#         Flag{53c82eba31f6d416f331de9162ebe997}                    #
##################################################################
barryallen@192.168.0.9's password:
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-38-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

57 packages can be updated.
0 updates are security updates.


*** System restart required ***

barryallen@skydogconctf2016:~$ whoami
barryallen
```

Una vez dentro de la máquina:

```
barryallen@skydogconctf2016:~$ ls -rlt
total 73016
-rw-r--r-- 1 barryallen barryallen       39 Oct 10 18:17 flag.txt
-rw-r--r-- 1 barryallen barryallen 74762682 Oct 10 18:29 security-system.data
barryallen@skydogconctf2016:~$ cat flag.txt
flag{bd2f6a1d5242c962a05619c56fa47ba6} = theflash
```

Y utilizando ya el último flag inicial:

Flag #8 Franks Lost His Mind or Maybe it's His Memory. He's Locked Himself Inside the Building. Find the Code to Unlock the Door Before He Gets Himself Killed!

Intentando leer el archivo security-system.data, observamos que parece tratarse de un binario. “strings” no está instalado en la máquina y poco más podemos hacer. Bajemos el archivo a nuestra máquina local para trabajar con él pues:

scp -P 22222 barryallen@192.168.0.9:security-system.data ~/Security/Skydog

Está claro que por el tamaño, por el nombre y tras buscar en google, se trata de un dump de memoria. Necesitaremos Volatility.

Tras extraer de nuevo el zip, evaluamos la imagen en crudo obtenida:

```
root@kali:~/Security/Skydog# volatility imageinfo -f security-system2.data
Volatility Foundation Volatility Framework 2.5
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : WinXPSP2x86, WinXPSP3x86 (Instantiated with WinXPSP2x86)
                     AS Layer1 : IA32PagedMemoryPae (Kernel AS)
                     AS Layer2 : FileAddressSpace (/root/Security/Skydog/security-system2.data)
                      PAE type : PAE
                           DTB : 0x33e000L
                          KDBG : 0x80545b60L
          Number of Processors : 1
     Image Type (Service Pack) : 3
                KPCR for CPU 0 : 0xffdff000L
             KUSER_SHARED_DATA : 0xffdf0000L
           Image date and time : 2016-10-10 22:00:50 UTC+0000
     Image local date and time : 2016-10-10 18:00:50 -0400
```

Esto ya nos dice el perfil que tenemos que utilizar. Ahora, para ver el historial de los comando introducidos en la consola, podemos utilizar:

consoles   Extract command history by scanning for _CONSOLE_INFORMATION

```
root@kali:~/Security/Skydog# volatility consoles --profile=WinXPSP2x86 -f security-system2.data
Volatility Foundation Volatility Framework 2.5
**************************************************
ConsoleProcess: csrss.exe Pid: 560
Console: 0x4f23b0 CommandHistorySize: 50
HistoryBufferCount: 1 HistoryBufferMax: 4
OriginalTitle: %SystemRoot%\system32\cmd.exe
Title: C:\WINDOWS\system32\cmd.exe
AttachedProcess: cmd.exe Pid: 1336 Handle: 0x2d4
----
CommandHistory: 0x10186f8 Application: cmd.exe Flags: Allocated, Reset
CommandCount: 2 LastAdded: 1 LastDisplayed: 1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x2d4
Cmd #0 at 0x1024400: cd Desktop
Cmd #1 at 0x4f2660: echo 66 6c 61 67 7b 38 34 31 64 64 33 64 62 32 39 62 30 66 62 62 64 38 39
63 37 62 35 62 65 37 36 38 63 64 63 38 31 7d > code.txt
----
Screen 0x4f2ab0 X:80 Y:300
```

Dump:
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\test>cd Desktop

C:\Documents and Settings\test\Desktop>echo 66 6c 61 67 7b 38 34 31 64 64 33 64 62 32 39 62 30 66 62 62 64 38 39 63 37 62 35 62 65 37 36 38 63 64 63 38 31 7d > code.txt

C:\Documents and Settings\test\Desktop>
****
ConsoleProcess: csrss.exe Pid: 560
Console: 0x1028488 CommandHistorySize: 50
HistoryBufferCount: 2 HistoryBufferMax: 4
OriginalTitle: ?OystemRoot%\system32\cmd.exe
Title:

Así pues, vemos que el código que buscamos (code.txt), está en hexadecimal. Podemos pasarlo a ascii en la consola de Python o mediante esta página:
http://www.rapidtables.com/convert/number/hex-to-ascii.htm

Y esto nos da el último flag:

flag{841dd3db29b0fbbd89c7b5be768cdc81}

841dd3db29b0fbbd89c7b5be768cdc81 MD5 : Two[space]little[space]mice