
HACK
ISTANBUL

İÇİNDEKİLER

STAGE 1

MAGIC

Magic adlı soruda bize 3 adet dosya veriyor.

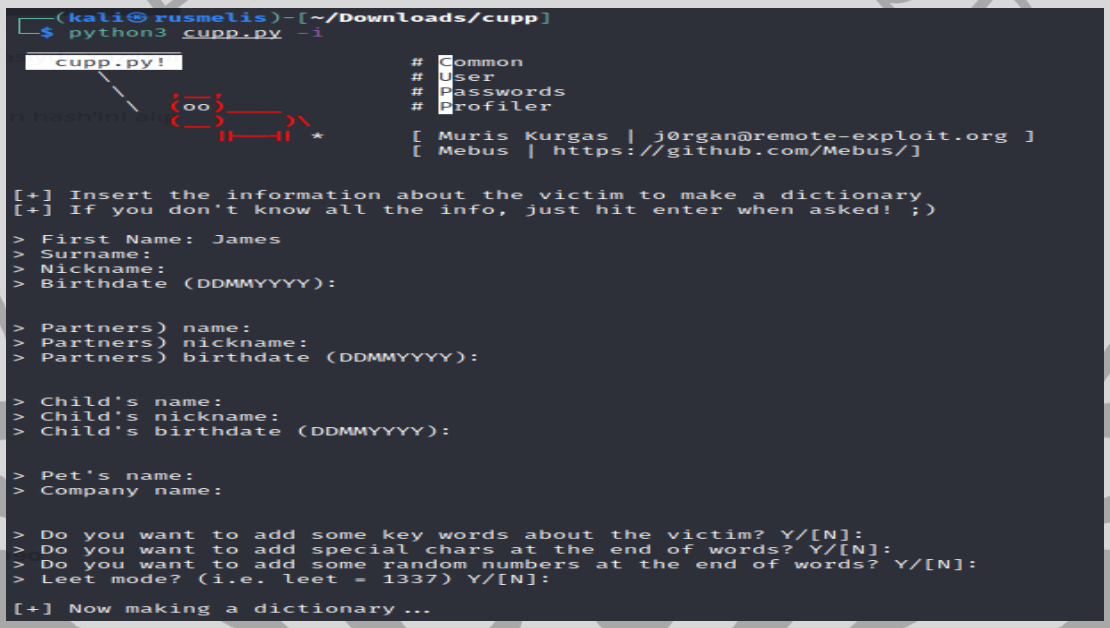
```
┌──(kali㉿rusmelis)-[~/Desktop/writeup/Magic]
└─$ ls
nothing_here.jpg  nothing_here.zip  only_the_worthy.png
```

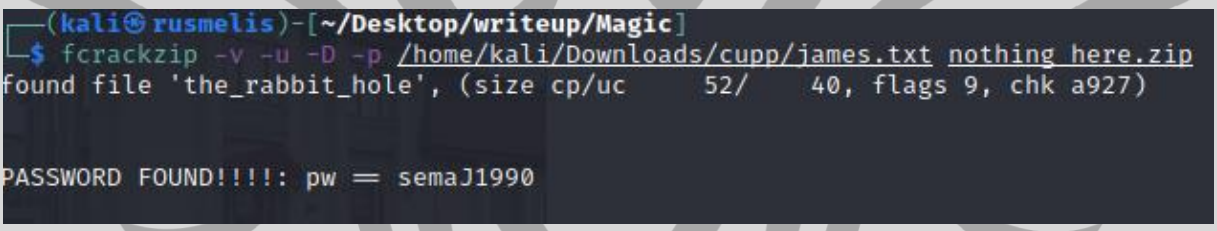
```
File: only_the_worthy.png                    ASCII Offset: 0x00000000 / 0x0004C12E (%00)
00000000  00 00 00 00  00 00 00 00   00 00 00 0D  49 48 44 52   ............IHDR
00000010  00 00 07 80  00 00 04 38   08 06 00 00  00 E8 D3 C1   .......8........
00000020  43 00 00 20  00 49 44 41   54 78 5E EC  9D 07 B8 15   C.. .IDATx^.....
00000030  D5 B9 BF 97  C9 4D AE BD   63 8D 5D 8C  B1 2B 06 8D   .....M..c.]..+..
00000040  25 18 EC 62  6F D8 50 B9   22 D8 0B 11  4B C4 C4 82   %..bo.P."...K...
00000050  88 5D 63 C5  82 8A 7A D5   18 8D 15 51  EC 8A 5D 14   .]c...z....Q..].
00000060  15 0D 96 8B  A2 28 F6 9E   68 D4 F8 7F  DE 75 FF 73   .....(..h....u.s
00000070  EE A6 1C 76  9B 99 3D 7B   9F F7 7B 9E  F3 1C E5 CC   ...v..={..{.....
00000080  5E B3 E6 5D  33 B3 D7 5A   BF AF CC B4  D4 52 4B FD   ^..]3..Z.....RK.
00000090  18 34 09 48  40 02 12 90   80 04 24 20  01 09 48 40   .4.H@.....$ ..H@
000000A0  02 12 90 80  04 24 20 01   09 48 40 02  12 90 80 04   .....$ ..H@.....
000000B0  24 20 01 09  48 40 02 12   68 7A 02 33  29 00 37 FD   $ ..H@..hz.3).7.
000000C0  18 7A 01 12  90 80 04 24   20 01 09 48  40 02 12 90   .z.....$ ..H@...
000000D0  80 04 24 20  01 09 48 40   02 12 90 80  04 24 20 01   ..$ ..H@.....$ .
000000E0  09 48 40 02  12 90 80 04   22 01 05 60  6F 04 09 48   .H@.....".."`o..H
000000F0  40 02 12 90  80 04 24 20   01 09 48 40  02 12 90 80   @.....$ ..H@....
00000100  04 24 20 01  09 48 40 02   12 90 80 04  24 20 01 09   .$ ..H@.....$ ..
00000110  48 40 02 2D  42 40 01 B8   45 06 D2 CB  90 80 04 24   H@.-B@..E......$
00000120  20 01 09 48  40 02 12 90   80 04 24 20  01 09 48 40   ..H@.....$ ..H@
00000130  02 12 90 80  04 24 20 01   09 48 40 02  12 90 80 04   .....$ ..H@.....
00000140  24 A0 00 EC  3D 20 01 09   48 40 02 12  90 80 04 24   $...= ..H@.....$
00000150  20 01 09 48  40 02 12 90   80 04 24 20  01 09 48 40   ..H@.....$ ..H@
00000160  02 12 90 80  04 24 20 01   09 48 A0 45  08 28 00 B7   .....$ ..H.E.(..
00000170  C8 40 7A 19  12 90 80 04   24 20 01 09  48 40 02 12   .@z.....$ ..H@..
00000180  90 80 04 24  20 01 09 48   40 02 12 90  80 04 24 20   ... $ ..H@.....$
00000190  01 09 48 40  02 12 90 80   04 14 80 BD  07 24 20 01   ..H@.........$ .
000001A0  09 48 40 02  12 90 80 04   24 20 01 09  48 40 02 12   .H@.....$ ..H@..
000001B0  90 80 04 24  20 01 09 48   40 02 12 90  80 04 24 20   ... $ ..H@.....$
000001C0  01 09 B4 08  01 05 E0 16   19 48 2F 43  02 12 90 80   .........H/C....
000001D0  04 24 20 01  09 48 40 02   12 90 80 04  24 20 01 09   .$ ..H@.....$ ..
000001E0  48 40 02 12  90 80 04 24   20 01 09 48  40 02 12 90   H@.....$ ..H@ ...
000001F0  80 02 B0 F7  80 04 24 20   01 09 48 40  02 12 90 80   ......$ ..H@....
00000200  04 24 20 01  09 48 40 02   12 90 80 04  24 20 01 09   .$ ..H@.....$ ..
00000210  48 40 02 12  90 80 04 24   20 81 16 21  A0 00 DC 22   H@.....$ ..! ..."
00000220  03 E9 65 48  40 02 12 90   80 04 24 20  01 09 48 40   ..eH@.....$ ..H@
00000230  02 12 90 80  04 24 20 01   09 48 40 02  12 90 80 04   .....$ ..H@.....
00000240  24 20 01 09  48 40 02 12   50 00 F6 1E  90 80 04 24   $ ..H@..P.......$
00000250  20 01 09 48  40 02 12 90   80 04 24 20  01 09 48 40   ..H@.....$ ..H@
00000260  02 12 90 80  04 24 20 01   09 48 40 02  12 90 80 04   .....$ ..H@.....
00000270  24 D0 22 04  14 80 5B 64   20 BD 0C 09  48 40 02 12   $." ...[d  ...H@..
00000280  90 80 04 24  20 01 09 48   40 02 12 90  80 04 24 20   ...$ ..H@.....$
00000290  01 09 48 40  02 12 90 80   04 24 20 01  09 48 40 02   ..H@.....$ ..H@.
^G Help  ^C Exit (No Save)  ^T goTo Offset  ^X Exit and Save  ^W Search
```

Dosyaları incelediğimizde only_the_worthy.png adlı dosyanın headerının png formatına uymadığını görüp png'ye çeviriyoruz ve bize alttaki fotoğrafı veriyor.

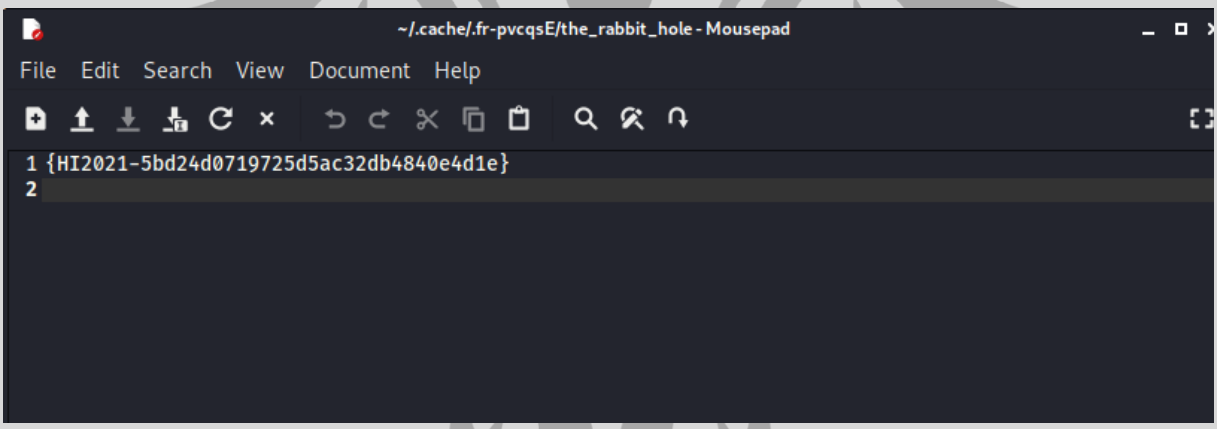
89 50 4E 47 0D 0A 1A 0A – PNG File Signature

Yukarıdaki fotoğrafta CUPP yazısını özellikle gösteriyor ve internette araştırdığımızda bunun bir wordlist oluşturma aracı olduğunu görüyoruz. Kullanacağımız bilgiler ise yukarıda fotoğrafta göründüğü gibi "james" olacak.





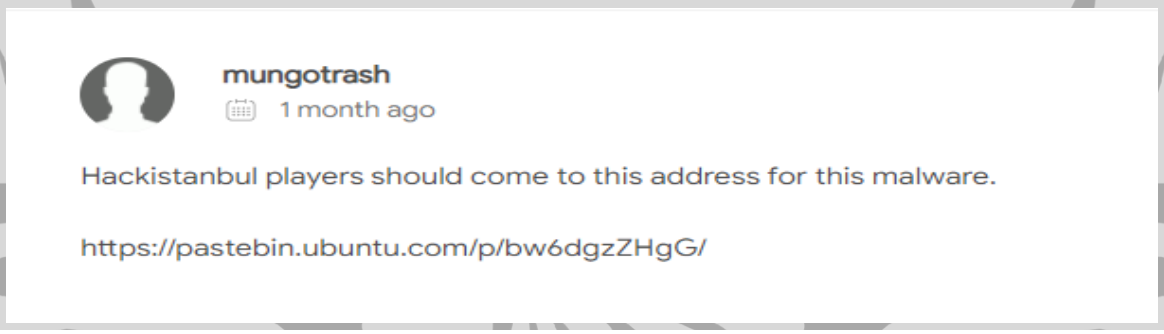
Yukarıda gördüğünüz gibi wordlist oluşturduk ve fcrackzip aracını kullanarak bize verilen zip dosyasının şifresini bulduk. Ve zip dosyasını açtığımızda flag karşımıza çıkıyor.

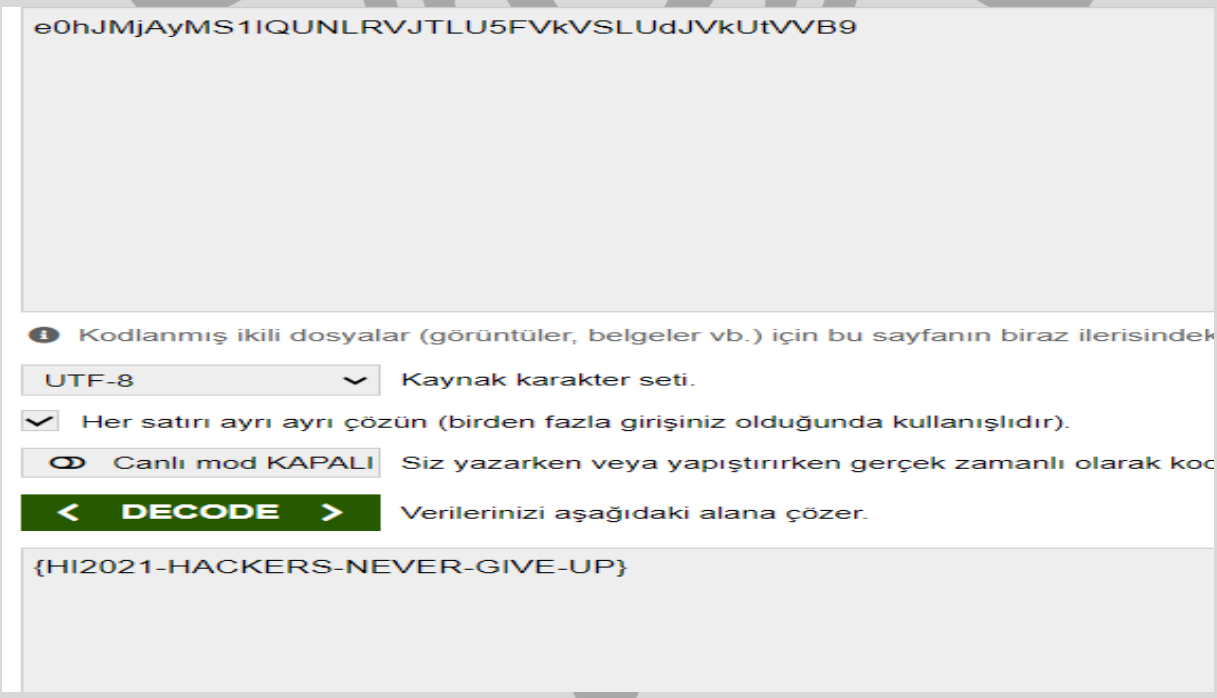


DETAİLED_SEARCH

Dosyada önümüze anlamsız bir karakter dizisi “7f7d1069ca8a852c1c8eb36e1d988fe6a9c17ecb8eff1f66fc5ebfeb5418723a” çıkıyor. Bunu https://www.virustotal.com/gui/home/search ile aradığımızda toplum kısmında bir ipucu çıkıyor ve o dosyaya ulaştığımızda yine bir şifrelenmiş veri ile karşılaşıyoruz.
“e0hJMjAyMS1IQUNLRVJTLU5FVkVSLUdJVkUtVVB9” base64 ile çözdüğümüzde flag karşımıza çıkıyor .

{HI2021-HACKERS-NEVER-GIVE-UP}





LİSTEN_WELL

Dinlediğimizde mors alfabesi olduğunu anladık bu yüzden https://morsecode.world/international/decoder/audio-decoder sayfasına sesi yükledik ve flag direkt karşımıza çıkıyor: *H ben2021-HACKERS-ARE-EVERYWHERE*

WHERE AM I

Soruda görseldeki uçağın hangi havaalanına indiğini soruyordu. Görseli arattığımızda uçağın Şanlıurfa Gap Havaalanına iniş yaptığını gördük.

BAYRAK: {HI2021-SANLIURFA-GAP-HAVALIMANI}

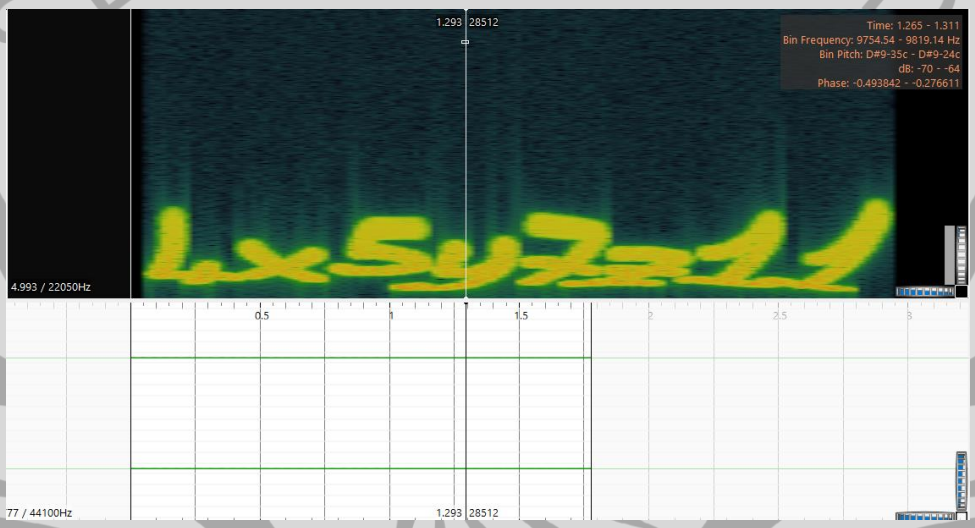
VAİL.TXT

Dosya içine baktığımızda bunun mors alfabesi ile yazıldığını anladık ve https://morsedecoder.com/tr/ sayfasına baktık.

Flag: *HI2021-CAUGHTME*

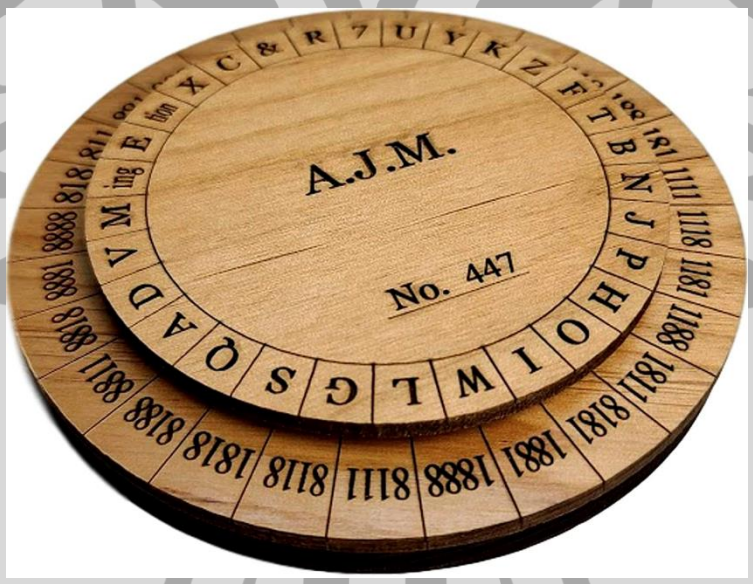
LİSTEN

Verilen .wav uzantılı dosyayı biraz inceledikten sonra spectrogram ekledik ve zip dosyasının şifresini gördük. Zip'in içinde ise flagi bulduk.



BAYRAK: **{HI2021-8ded3ddcf29e7434abc28b269f1ebe}**

GUERRA



Soruda bilgi olarak şifreleme disk görseli ve IAMALBERTI yazısı verilmişti. Görseli biraz araştırdıktan sonra Alberti Cipher olduğunu gördük. Alberti's Disk Encoder ile IAMALBERTI'yi encode ettikten sonra bayrağa ulaştık.

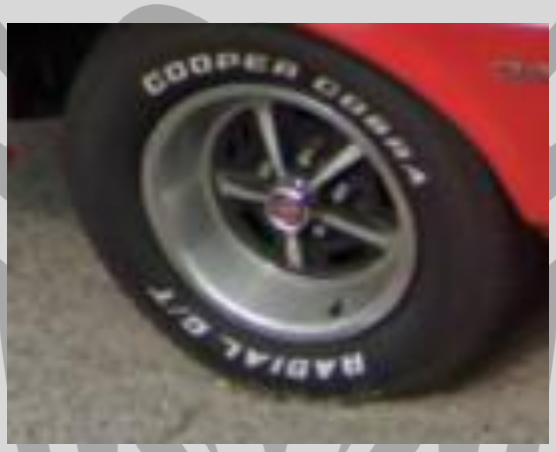
BAYRAK: {HI2021-FNCZBGSE}

WALKER

Bu soruda 5 adet dosya verildi ve dosyaları incelediğimizde bizden bir suçluyu yakalamak için yardım istendiğini gördük.

```
┌──(kali㉿rusmelis)-[~/Desktop/writeup/Walker]
└─$ ls
classified  daniel_wright.pdf  doc  loc  private
```

Bize verilen **classified** adlı dosyada Daniel Wright adlı suçlu hakkında bilgi veriliyor. Bu kişinin gizli isminin C.Cobra olduğunu söylüyor ve bize son görüldüğü yerleri söyledikleri için Google Maps'i kullanarak biraz geziniyoruz. Bize tarif edilen arabayı buluyoruz ve tekerlerinde suçlunun ismini nereden bulduğunu görüyoruz.



Şifreli dosyada şifreyi deneyerek şifrenin "coopercobra" olduğunu buluyoruz. **Doc** dosyasını decrypt ettiğimizde ise bayrağa ulaşıyoruz.

```
┌──(kali㉿rusmelis)-[~/Desktop/writeup/Walker]
└─$ openssl rsautl -decrypt -inkey private -in doc -out res
Enter pass phrase for private:

┌──(kali㉿rusmelis)-[~/Desktop/writeup/Walker]
└─$ ls
classified  daniel_wright.pdf  doc  loc  private  res

┌──(kali㉿rusmelis)-[~/Desktop/writeup/Walker]
└─$ strings res
Hi James,
Thanks for transfer in $. Now, let me send the password for account.
{HI2021-a6f7b874ea69329372ad75353314d7}
It was a pleasure working with you.
Daniel W.
```

STAGE 2

CAN YOU READIT

Soruda barkod içeren bir sürücü kartı verilmişti. Görsele online barkod okuyucu ile baktığımızda bayrağa ulaştık.

BAYRAK: **{HI2021-7a56be269cb3adde7ed81b4d00d50d}**

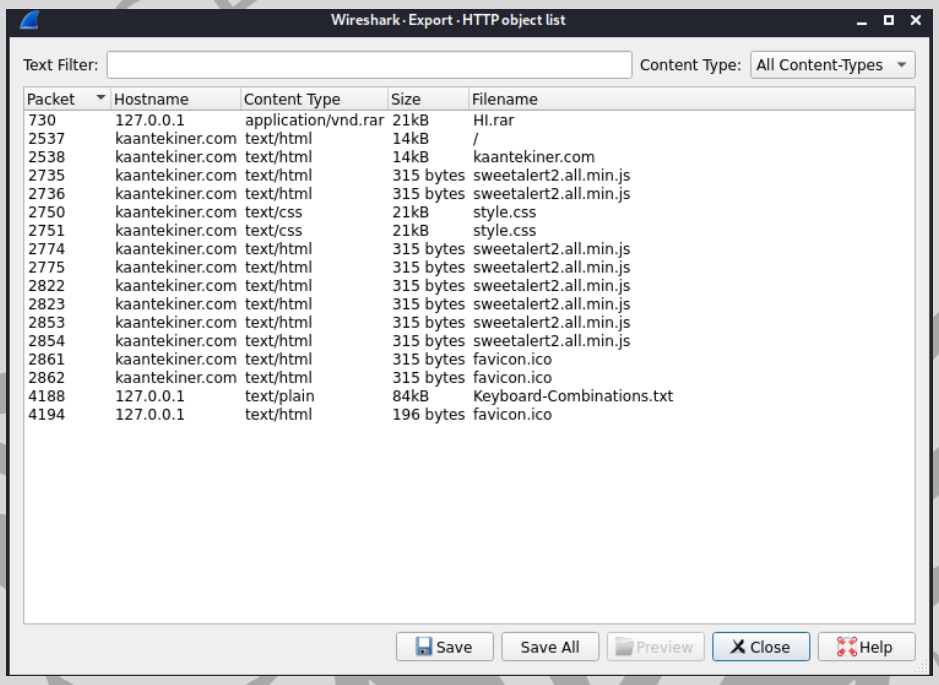
H3X.TXT

Bu soruda bize verilen .txt uzantılı dosyayı incelediğimizde headerda, aslında bir png dosyası olduğunu görüyoruz. Hexlerini değiştirdiğimizde ise flagi direkt olarak bize veriyor.

{HI2021-7a56be269cb3adde7ed81b4d00d50d}

WHATISPASSWORD

Bu soruda bir pcap dosyası verildi. Dosyayı incelediğimizde HI.rar ve Keyboard-Combinations.txt dosyalarını gördük.



Dosyaları incelemek için indirdiğimizde .rar uzantılı dosyanın şifreli olduğunu gördük. John aracını kullanarak .rar uzantılı dosyayı verilen wordlisti kullanarak açtık. Ve içinden .jpg uzantılı bir dosya çıktı. Bu fotoğrafı incelediğimizde flagi içinde bulduk.

```
Text Layer Name                 : {HI2021-fcc637036c3c11f699c22c243dd82a}
Text Layer Text                 : {HI2021-xxxxxxxxxxxxxxxxxxxxxxxx}
```

XSS-Brute-Logic

Bu soruda bize bir .zip uzantılı bir dosya veriyor. Açtığımızda ise içinden dosya ismi ile aynı bir .png uzantılı dosya ve .git dizini çıktı. Resmi inceledik fakat bir şeyler çıkartamadık. Git loglarına baktığımızda ise bir github hesabı olduğunu gördük.

```
hako@66616b6f: /mnt/d/LinuxDownloads/XSS-BruteLogic
hako@66616b6f:/mnt/d/LinuxDownloads/XSS-BruteLogic$ ls
XSS-BruteLogic.txt.png
hako@66616b6f:/mnt/d/LinuxDownloads/XSS-BruteLogic$ ls -alh
total 1.0K
drwxrwxrwx 1 hako hako 512 Aug 13 00:47 .
drwxrwxrwx 1 hako hako 512 Aug 13 00:47 ..
drwxrwxrwx 1 hako hako 512 Jun 25 17:07 .git
-rwxrwxrwx 1 hako hako 621 Jun 20 19:31 XSS-BruteLogic.txt.png
hako@66616b6f:/mnt/d/LinuxDownloads/XSS-BruteLogic$
```

```
hako@66616b6f: /mnt/d/LinuxDownloads/XSS-BruteLogic/.git/logs
hako@66616b6f:/mnt/d/LinuxDownloads/XSS-BruteLogic/.git/logs$ ls
HEAD  refs
hako@66616b6f:/mnt/d/LinuxDownloads/XSS-BruteLogic/.git/logs$ strings HEAD
0000000000000000000000000000000000000000 2b42a92ca4caf19c11ab074f72c7e5105f2654ec effective <effective@tulpar.local>
4204010 +0300    clone: from https://github.com/Effective1212/Steganography.git
hako@66616b6f:/mnt/d/LinuxDownloads/XSS-BruteLogic/.git/logs$
```

Githubda verilen open-cv kütüphanesi ile yazılmış steg.py scriptini incelediğimizde bize verilen resimin encrypt edildiğini ve scriptin bir decoder olduğunu anladık. İçinde bulunan decrpyt fonksiyonunu bir brute force aracı gibi kullanarak soruyu çözdük.

Wordlist: https://github.com/danielmiessler/SecLists/blob/master/Fuzzing/XSS/XSS-BruteLogic.txt

NT OR LM

Bu soruda bize 1 gb boyutunda bir image dosyası veriyor. Direkt Volatility-Master toolu ile incelemeye başladık. Dosyanın imageinfo'suna bakarak profili bulduk.

```
C:\Users\Hakan>C:\Users\Hakan\Desktop\volatility_2.6_win64_standalone\volatility_2.6_win64_standalone\volatility_2.6_win
64_standalone.exe -f "C:\Users\Hakan\Desktop\HI2021\HI2021\NT_0R_LM (1).ram" imageinfo
Volatility Foundation Volatility Framework 2.6
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : Win7SP1x64, Win7SP0x64, Win2008R2SP0x64, Win2008R2SP1x64_23418, Win2008R2SP1x64, Win7SP
1x64_23418
                     AS Layer1 : WindowsAMD64PagedMemory (Kernel AS)
                     AS Layer2 : FileAddressSpace (C:\Users\Hakan\Desktop\HI2021\HI2021\NT_0R_LM (1).ram)
                      PAE type : No PAE
                           DTB : 0x187000L
                          KDBG : 0xf800029f30a0L
          Number of Processors : 1
     Image Type (Service Pack) : 1
                KPCR for CPU 0 : 0xfffff800029f4d00L
             KUSER_SHARED_DATA : 0xfffff78000000000L
           Image date and time : 2021-06-20 19:28:04 UTC+0000
     Image local date and time : 2021-06-20 22:28:04 +0300

C:\Users\Hakan>C:\Users\Hakan\Desktop\volatility_2.6_win64_standalone\volatility_2.6_win64_standalone\volatility_2.6_win
64_standalone.exe -f "C:\Users\Hakan\Desktop\HI2021\HI2021\NT_0R_LM (1).ram" --profile=Win7SP1x64
```

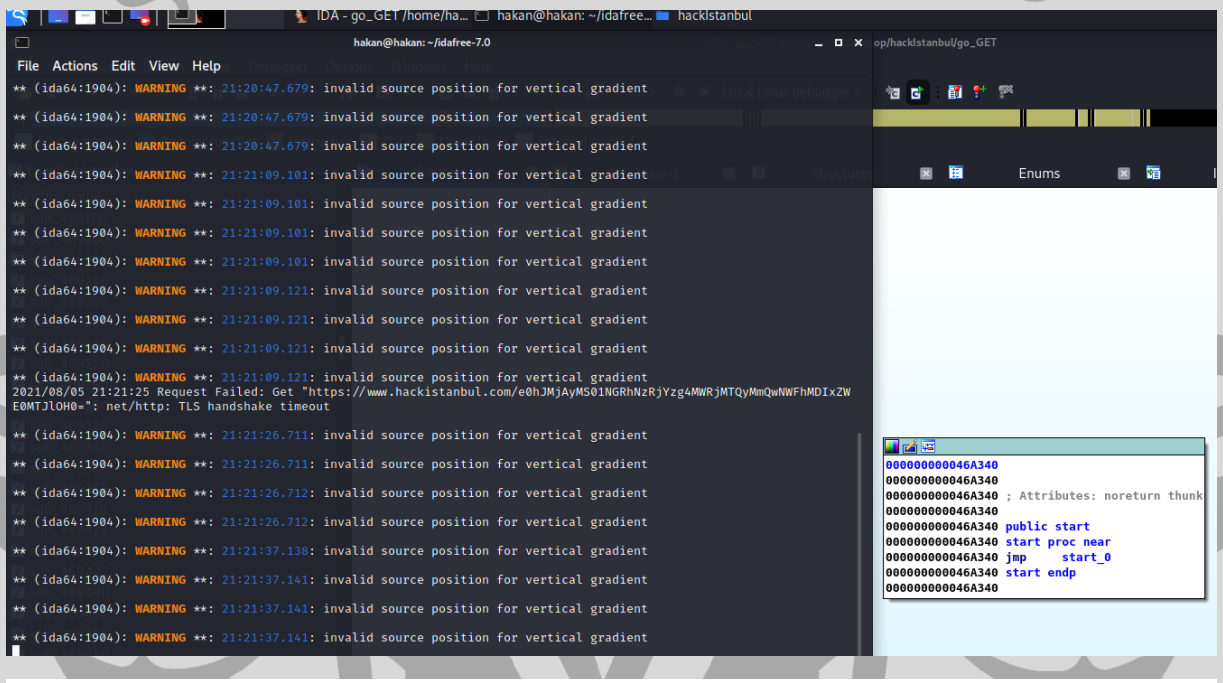
Daha sonra bizden NT ya da LM şifrelemesi ile alakalı bir şey bulmamızı istediği için direkt olarak hashlere baktık.

```
C:\Users\Hakan>C:\Users\Hakan\Desktop\volatility_2.6_win64_standalone\volatility_2.6_win64_standalone\volatility_2.6_win
64_standalone.exe -f "C:\Users\Hakan\Desktop\HI2021\HI2021\NT_0R_LM (1).ram" --profile=Win7SP1x64 hashdump
Volatility Foundation Volatility Framework 2.6
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
effective:1000:aad3b435b51404eeaad3b435b51404ee:e45a314c664d40a227f9540121d1a29d:::
HomeGroupUser$:1002:aad3b435b51404eeaad3b435b51404ee:0609c7cb7ab9ba547c23fd7fe1aa12b8:::
hackistanbul:1003:aad3b435b51404eeaad3b435b51404ee:049c5449f08de05c3f06d2a22dd0b593:::
```

Hashi bulduktan sonra tek tek LM ve NT'yi denedik ve bayrak LM çıktı.

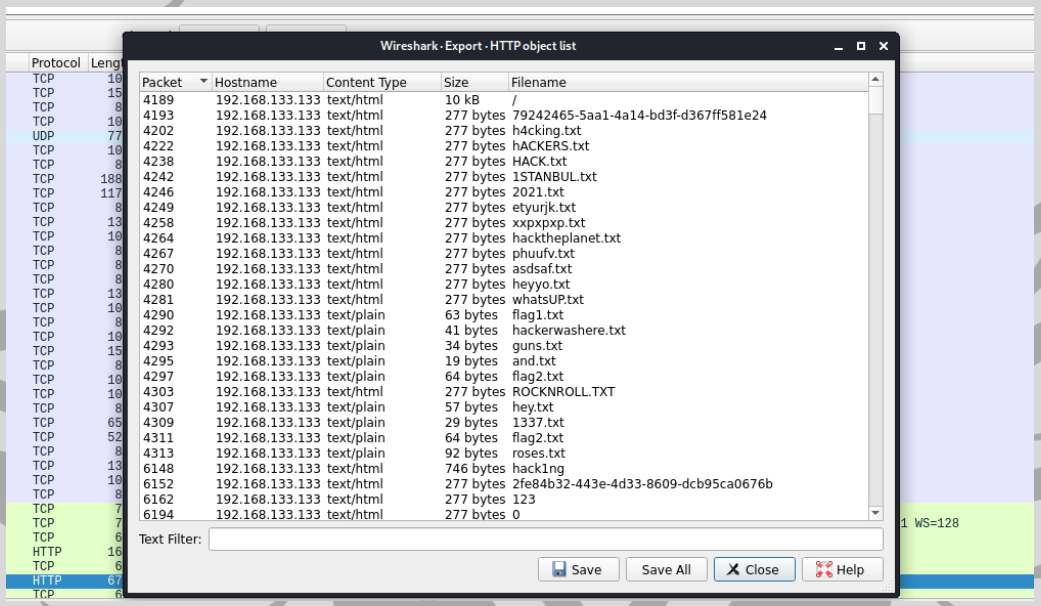
GOGET

Bu soruda verilen dosyayı incelediğimizde yürütülebilir bir dosya olduğunu görüyoruz. Yürütülebilir dosyayı incelediğimizde bir istek attığını görüyoruz. Web sitesinin altındaki dizinin base64 ile encode edildiğini düşünüp decode ettiğimizde bayrağımızı bulmuş oluyoruz.



ROCKN ROLL

Soruda bize bir .pcap uzantılı dosya veriyordu. Wireshark incelediğimizde .txt uzantılı dosyalar ile karşılaştık ve flagin 3 kısıma ayrıldığını farkettik.




```
toor@yasemin:~/Downloads$ cat guns.txt

1/3

PNEESMRQGIYS2U3XGMZXIXY=

toor@yasemin:~/Downloads$ cat and.txt

2/3

Q2gxbERfT18=
toor@yasemin:~/Downloads$ cat roses.txt
3/3

5b79c0b5029b7e1bf25767d93bb07321


ENJ0Y! https://www.youtube.com/watch?v=1w7OgIMMRc4
toor@yasemin:~/Downloads$
```

Sırası ile Base32 Base64 ve RC2 ile decode ettiğimizde bayrağa ulaşıyoruz

BAYRAK: **{HI2021-Sw33t_Ch1lD_O_M1n3_R0CK}**

ARCFOUR

Soruda oynanmış bir QR kod verilmişti. Online araçlarla düzelterek kodu okuttuk ve şifrelenmiş bir metin ve şifresi ile karşılaştık.

Pjb2PFGwOH6QSej6HZhehqTV1AXre9FrXqaattsnWZQ6h0+aKC/qzqs=

Pass:CBCHackIstanbul

RC4 ile decode ettikten sonra bayrağa ulaştık.

BAYRAK: **{HI2021-62f170fb07fdbb79ceb7147101406eb8}**

STAGE 3

ZİPPED FLAG

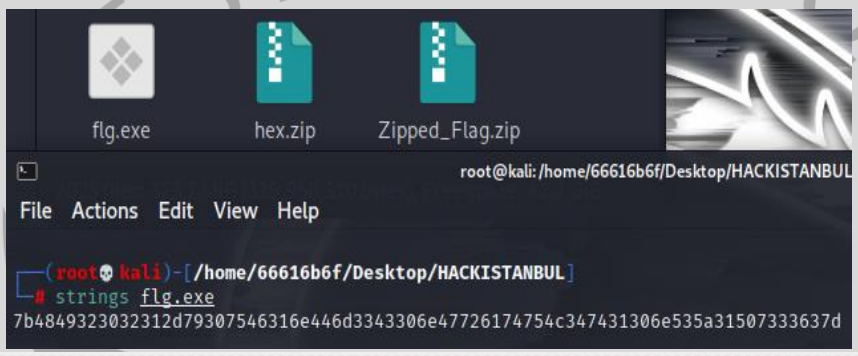
Bu soruda .zip uzantılı dosya veriliyor. İnceledikten sonra içinde çok fazla zip dosyası olduğunu gördük ve bir script yazarak 2021 tane zip dosyasını çıkarıyoruz.

```
┌──(root💀kali)-[/home/66616b6f/Desktop/HACKISTANBUL]
└─# unzip_all() {
    zipfile="$1"
    next_zipfile="$(unzip -Z1 "$zipfile" | head -n1)"
    if echo "$next_zipfile" | grep ".zip$"; then
        unzip -P "${next_zipfile%%.*}" "$zipfile"
        unzip_all "$next_zipfile"
    fi
}
unzip_all "1000.zip"
```

Son dosyanın içinde hex.zip adlı bir dosya çıkıyordu.

```
Archive:  2020.zip
 extracting: 2021.zip
hex.zip
Archive:  2021.zip
 extracting: hex.zip
```

Hex.zip dosyasını açtığımızda flg.exe adlı dosya çıkıyor ve incelediğimizde içinde bir hex buluyoruz. Decrypt ettiğimizde ise flagi elde ediyoruz.



Hex String	{HI2021-y0uF1nDm3C0nGratuL4t10nSZ1Ps3c}

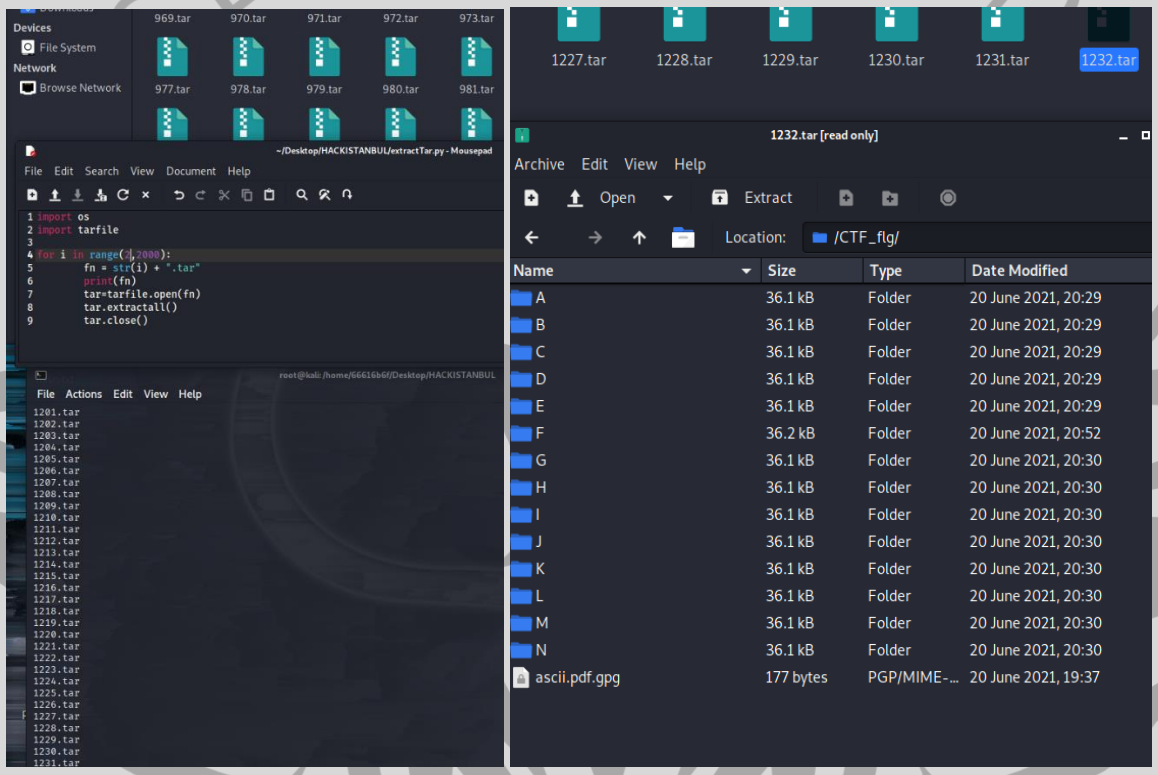
VULCAN

Bu soruyu steghide-stagcracker ikilisi ile çözdük. Stegcracker yardımıyla şifreyi bularak, steghide ile de içerisinden gizli dosyayı çıkardık. İçindeki kodu rot47 ile çözdükten sonra flagi bulduk:

BAYRAK :**{HI2021-81919ca17ded3ac3420d4c714d362d}**

B2B21

Bu soruda b2b21.tar adlı dosya veriliyor. İncelediğimizde içinde çok fazla .tar uzantılı dosya olduğunu görüyoruz. Bunu açmak için yine bir script kullandık.



Son dosyaya kadar açtıktan sonra karşımıza farklı bir sıkıştırılmış dosya geliyor. İçinde A'dan N'ye kadar sıralanmış dizinler ve ascii.pdf.gpg adlı dosya görüyoruz. F dizininde şifrelenmiş bir zip dosyası ve içinde flag.exe olduğunu görüyoruz. Birkaç denemeden sonra zipin kırılmayacağını anladık ve odağımızı ascii.pdf.gpg'ye verdik. ascii.pdf.gpg'yi john ile kırdıktan sonra bize bir ipucu verdi ve zipde denedik. Zip'i kırdıktan sonra bize bir binary verdi ve texte çevirdiğimizde bayrağımızı bulmuş olduk.

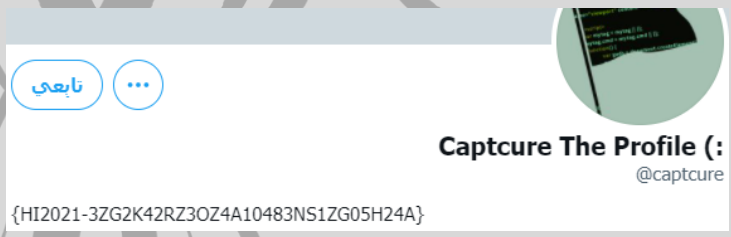
```
┌──(root💀kali)-[/home/…/Desktop/HACKISTANBUL/CTF_flg/F]
└─# gpg2john ascii.pdf.gpg > hash

File ascii.pdf.gpg
[gpg2john] MDC is missing, expect lots of false positives!

┌──(root💀kali)-[/home/…/Desktop/HACKISTANBUL/CTF_flg/F]
└─# john hash
Using default input encoding: UTF-8
Loaded 1 password hash (gpg, OpenPGP / GnuPG Secret Key [32/64])
Cost 1 (s2k-count) is 14155776 for all loaded hashes
Cost 2 (hash algorithm [1:MD5 2:SHA1 3:RIPEMD160 8:SHA256 9:SHA384 10:SHA512 11:SHA224]) is 2 for all loaded hashes
Cost 3 (cipher algorithm [1:IDEA 2:3DES 3:CAST5 4:Blowfish 7:AES128 8:AES192 9:AES256 10:Twofish 11:Camellia128 12:Camellia
192 13:Camellia256]) is 3 for all loaded hashes
Will run 4 OpenMP threads
Note: This format may emit false positives, so it will keep trying even after
finding a possible candidate.
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
princess         (?)

┌──(root💀kali)-[/home/…/Desktop/HACKISTANBUL/CTF_flg/F]
└─# gpg --decrypt ascii.pdf.gpg                                    1 ✕
gpg: CAST5.CFB encrypted data
gpg: encrypted with 1 passphrase
79 104 33 32 78 111 44 32 89 111 117 32 109 117 115 116 32 99 111 110 116 105 110 117 101 46 32 84 104 105 115 32 116 101 1
20 116 32 104 97 115 110 39 116 32 103 111 116 32 97 32 123 70 125 108 97 103 32 58 68 32 123 67 84 70 45 67 48 110 116 49
78 117 51 125 10

Oh! No, You must continue. This text hasn't got a {F}lag :D {CTF-C0nt1Nu3}
```

LOOK IN DETAIL

Bu soruda verilen resmin exiftool ile metadatasına baktığımızda başka bir resim adresi gördük.

```
toor@yasemin:~/Documents$ exiftool Look_in_Detail.jpg
ExifTool Version Number         : 12.16
File Name                       : Look_in_Detail.jpg
Directory                       : .
File Size                       : 53 KiB
File Modification Date/Time     : 2021:07:31 10:23:33+03:00
File Access Date/Time           : 2021:08:05 21:37:45+03:00
File Inode Change Date/Time     : 2021:07:31 10:34:31+03:00
File Permissions                : r-xr-xr-x
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Resolution Unit                 : inches
X Resolution                    : 96
Y Resolution                    : 96
XMP Toolkit                     : Image::ExifTool 12.27
Description                     : ibbDOTco/tMDTJxP
Comment                         : CREATOR: gd-jpeg v1.0 (using IJG JPEG v62)
Image Width                     : 735
Image Height                    : 491
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
```

Görseli Google Görsel Arama ile arattığımızda karşımıza bir tweet çıktı ve bayrağa ulaştık.



BAYRAK: **{HI2021-3ZG2K42RZ3OZ4A10483NS1ZG05H24A}**

NTLM_somethingelse

Bu soruda yine bize bir 1 gb image dosyası veriyor ve çözümü tamamen NT OR LM ile aynı.

```
C:\Users\Hakan>C:\Users\Hakan\Desktop\volatility_2.6_win64_standalone\volatility_2.6_win64_standalone\volatility_2.6_win
64_standalone.exe -f C:\Users\Hakan\Desktop\NTLM_or_somethingelse.ram imageinfo
Volatility Foundation Volatility Framework 2.6
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : Win7SP1x64, Win7SP0x64, Win2008R2SP0x64, Win2008R2SP1x64_23418, Win2008R2SP1x64, Win7SP
1x64_23418
                     AS Layer1 : WindowsAMD64PagedMemory (Kernel AS)
                     AS Layer2 : FileAddressSpace (C:\Users\Hakan\Desktop\NTLM_or_somethingelse.ram)
                      PAE type : No PAE
                           DTB : 0x187000L
                          KDBG : 0xf800029f50a0L
          Number of Processors : 1
     Image Type (Service Pack) : 1
                KPCR for CPU 0 : 0xfffff800029f6d00L
             KUSER_SHARED_DATA : 0xfffff78000000000L
           Image date and time : 2021-06-21 14:40:33 UTC+0000
     Image local date and time : 2021-06-21 17:40:33 +0300

C:\Users\Hakan>C:\Users\Hakan\Desktop\volatility_2.6_win64_standalone\volatility_2.6_win64_standalone\volatility_2.6_win
64_standalone.exe -f C:\Users\Hakan\Desktop\NTLM_or_somethingelse.ram --profile=Win7SP1x64
```

```
C:\Users\Hakan>C:\Users\Hakan\Desktop\volatility_2.6_win64_standalone\volatility_2.6_win64_standalone\volatility_2.6_win
64_standalone.exe -f C:\Users\Hakan\Desktop\NTLM_or_somethingelse.ram --profile=Win7SP1x64 hashdump
Volatility Foundation Volatility Framework 2.6
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
effective:1000:aad3b435b51404eeaad3b435b51404ee:e45a314c664d40a227f9540121d1a29d:::
HomeGroupUser$:1002:aad3b435b51404eeaad3b435b51404ee:0609c7cb7ab9ba547c23fd7fe1aa12b8:::
hackistanbul:1003:aad3b435b51404eeaad3b435b51404ee:049c5449f08de05c3f06d2a22dd0b593:::
```

STIPES

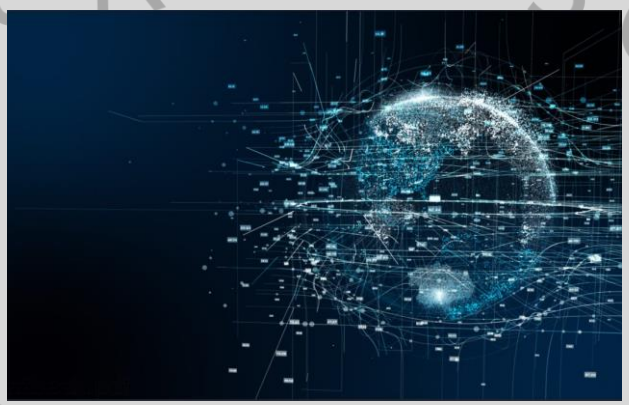
Bu soruda yürütülebilir dosya veriliyor. Dosyayı çalıştırdığımızda Hex veriyor decrypt ettiğimizde flagi buluyoruz.

```
┌──(kali㉿rusmelis)-[~/Desktop/writeup]
└─$ chmod +rwx stipes                                                    126 ×
```

```
┌──(kali㉿rusmelis)-[~/Desktop/writeup]
└─$ ./stipes
7b4849323032312d383865323832356436333836646362656366306633303039623365356261373035383730316334337dd41d8cd98f00b204e9800998ecf8427e
```

{HI2021-88e2825d6386dcbecf0f3009b3e5ba7058701c43}

Is_it_empty



Bu sorumuzda bize bir fotoğraf veriliyor inceliyoruz ve “atbash.pdf” yazısı haricinde bir şey bulamıyoruz. Stegcracker adlı aracı kullanarak brute force uyguluyoruz ve bir şifre buluyoruz.

```
┌──(kali㉿rusmelis)-[~/Desktop/writeup]
└─$ stegcracker is_it_empty.jpeg /home/kali/Downloads/rockyou.txt                1 ✘
StegCracker 2.1.0 - (https://github.com/Paradoxis/StegCracker)
Copyright (c) 2021 - Luke Paris (Paradoxis)

StegCracker has been retired following the release of StegSeek, which
will blast through the rockyou.txt wordlist within 1.9 second as opposed
to StegCracker which takes ~5 hours.

StegSeek can be found at: https://github.com/RickdeJager/stegseek

Counting lines in wordlist..
Attacking file 'is_it_empty.jpeg' with wordlist '/home/kali/Downloads/rockyou.txt'..
Successfully cracked file with password: california
Tried 982 passwords
Your file has been written to: is_it_empty.jpeg.out
california
```

Bu şifreyle steghide aracını kullanarak içinden “b1.tar.gz” adlı bir dosya çıkarıyoruz.

```
┌──(kali㉿rusmelis)-[~/Desktop/writeup]
└─$ steghide info is_it_empty.jpeg
"is_it_empty.jpeg":
  format: jpeg
  capacity: 22.7 KB
Try to get information about embedded data ? (y/n) y
Enter passphrase:
  embedded file "b1.tar.gz":
    size: 9.5 KB
    encrypted: rijndael-128, cbc
    compressed: yes

┌──(kali㉿rusmelis)-[~/Desktop/writeup]
└─$ steghide --extract -sf is_it_empty.jpeg
Enter passphrase:
wrote extracted data to "b1.tar.gz".
```

Stringlerine baktığımızda bunun bir pdf olduğunu görüyoruz ve bize verilen fotoğrafın içindeki ipucunu kullanarak “b1.tar.gz” adlı dosyanın headerını değiştiriyoruz ve şifreli bir pdf dosyası elde ediyoruz.

25 50 44 46 2D	%PDF-	0	pdf	PDF document[21]

Pdf dosyasını kırabilmek için john aracını kullanıyoruz ve pdf dosyasının şifresini buluyoruz.

```
┌──(kali㉿rusmelis)-[/usr/share/john]
└─$ ./pdf2john.pl /home/kali/Desktop/writeup/b1.pdf > /home/kali/Desktop/writeup/pdf.hash
```

```
┌──(root💀rusmelis)-[/usr/share]
└─# john /home/kali/Desktop/writeup/pdf.hash                                    126 ×
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 1 password hash (PDF [MD5 SHA2 RC4/AES 32/64])
Cost 1 (revision) is 3 for all loaded hashes
Will run 4 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 3 candidates buffered for the current salt, minimum 8 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
blessed1         (/home/kali/Desktop/writeup/b1.pdf)
1g 0:00:00:10 DONE 2/3 (2021-08-09 12:22) 0.09496g/s 5846p/s 5846c/s 5846C/s jojo1..bonita1
Use the "--show --format=PDF" options to display all of the cracked passwords reliably
Session completed
```

Şifreyi girdiğimizde bize flagi veriyor.

{SR2021-692uyw4x4054u70z2648076z02850y}

SesamosSQD Team

Hakan SOYSAL(C)

https://www.linkedin.com/in/hakansoysal/

Yasemin BOZACI

https://tr.linkedin.com/in/yasemin-bozacı-a44621190

Hüseyin Selim SÜRMELİHİNDİ

https://tr.linkedin.com/in/hüseyin-selim-sürmelihindi-bb312a1b8

Rümeysa ERDOĞAN

https://tr.linkedin.com/in/rumeysa-erdoğan-8601791b3