

## A case history: CTF Necromancer - Parte 2

**Author** : Vincenzo Digilio

**Date** : 18 Maggio 2020



*L'articolo tratta la seconda parte della CTF Necromancer, tenuta come lezione all'Università di Perugia in occasione dell'evento CyberChallenge: <https://cyberchallenge.it/>  
La prima parte è stata pubblicata [qui](#).*

Seguendo i corvi in formazione all'orizzonte, al termine della prima parte eravamo giunti **sull'orlo dell'abisso**, circondati da nient'altro che il silenzio e le tenebre di un nuovo enigma.

Sulla sponda opposta... la grotta del Necromante.

### **FLAG – 03.**

A una fugace occhiata, premendo F12, non scorsi nulla di particolare nella pagina web (Figura 15).

The necromancer looks towards you with hollow eyes which can only be described as death.  
He smirks in your direction, and suddenly a bright light momentarily blinds you.  
The silence is broken by a blood curdling screech of a thousand birds, followed by the necromancers laughs fading as he decends into the cave!  
The crows break their formation, some flying aimlessly in the air; others now motionless upon the ground.  
The cave is now protected by a gaseous blue haze, and an organised pile of feathers lay before you.

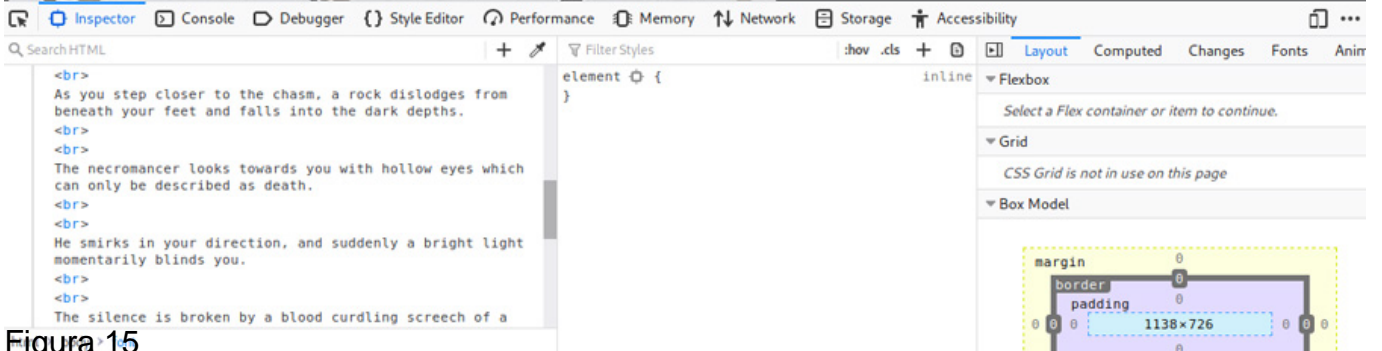


Figura 15

Ciò che invece attirò la mia attenzione fu l'immagine: pileoffeathers.jpg (Figura 16).

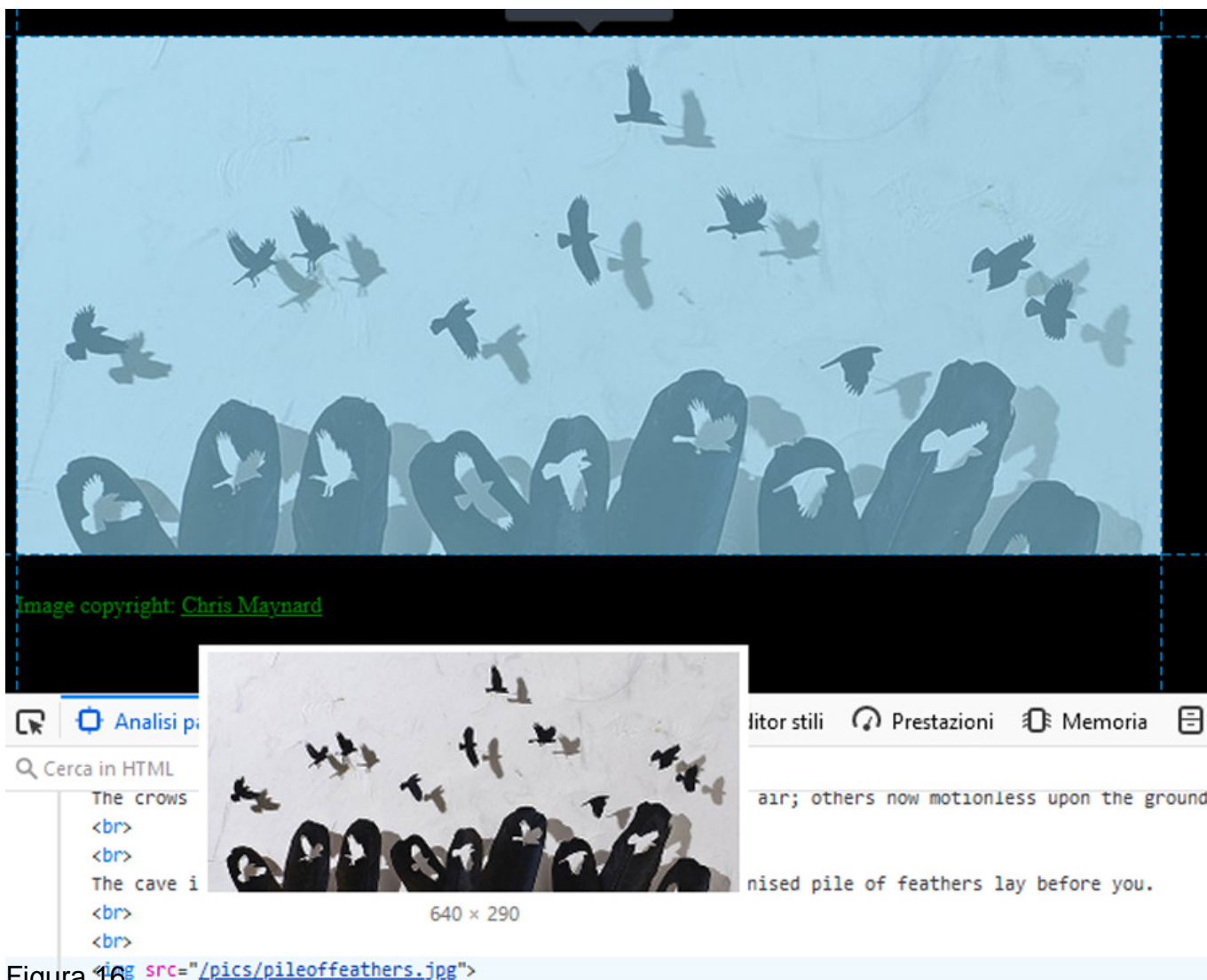


Figura 16

Utilizzerai il comando `wget` per scaricare l'immagine:

- `wget` : comando
- `168.178.37/pics/pileoffeathers.jpg` : il percorso all'immagine.

```

root@kali:/home/kali# wget 192.168.178.37/pics/pileoffeathers.jpg
--2020-04-20 10:53:26-- http://192.168.178.37/pics/pileoffeathers.jpg
Connecting to 192.168.178.37:80 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 37289 (36K) [image/jpeg]
Saving to: 'pileoffeathers.jpg'

pileoffeathers.jpg 100%[=====] 36.42K --.-KB/s in 0.002s
2020-04-20 10:53:26 (20.9 MB/s) - 'pileoffeathers.jpg' saved [37289/37289]

```

Una volta scaricata, effettuai l'analisi (Figura 17).

```
root@kali:/home/kali/Desktop/inventario# binwalk pileoffeathers.jpg
DECIMAL      HEXADECIMAL  DESCRIPTION
-----
0            0x0         JPEG image data, EXIF standard
12          0xC         TIFF image data, little-endian offset of first image directory: 8
36994       0x9082      Zip archive data, at least v2.0 to extract, compressed size: 121, uncompressed size: 125, name:
feathers.txt
37267       0x9193      End of Zip archive, footer length: 22
```

Utilizzai il *tool* binwalk richiamandolo con l'omonimo comando: esso permette di analizzare file binari per cercare file o codice eseguibile, inoculati all'interno. Difatti, questo *tool* trova il suo uso principale nell'estrazione del contenuto di immagini firmware.

- binwalk : comando
- jpg : nome dell'immagine da analizzare

Il *tool* rivelò che all'interno dell'immagine era stato nascosto un archivio .zip.

Sempre utilizzando binwalk procedetti all'estrazione del .zip (Figura 18):

```
root@kali:/home/kali/Desktop/inventario# binwalk --dd='.*' pileoffeathers.jpg
DECIMAL      HEXADECIMAL  DESCRIPTION
-----
0            0x0         JPEG image data, EXIF standard
12          0xC         TIFF image data, little-endian offset of first image directory: 8
36994       0x9082      Zip archive data, at least v2.0 to extract, compressed size: 121, uncompressed size: 125, name: feathers.txt
37267       0x9193      End of Zip archive, footer length: 22
```

- binwalk : comando
- --dd='.\*' : estrai dall'immagine i file con una determinata estensione (l'asterisco identificava "ogni tipo di...")
- jpg : nome dell'immagine da cui estrarre

L'*output* del comando fu una lista di quattro file (Figura 19). Utilizzando il comando `file *` (per listare tutti i tipi di file in una directory) mi si presentò l'archivio chiamato 9082.

```
root@kali:/home/kali/Desktop/inventario/_pileoffeathers.jpg-0.extracted# file *
0:      JPEG image data, Exif standard: [TIFF image data, little-endian, direntries=1]
9082:   Zip archive data, at least v2.0 to extract
9193:   Zip archive data (empty)
9727:   TIFF image data, little-endian, direntries=0
```

Non mi rimaneva che procedere all'*"unzip"* del file (Figura 20).

```
root@kali:/home/kali/Desktop/inventario/_pileoffeathers.jpg-0.extracted# unzip 9082
Archive: 9082
inflating: feathers.txt
```

- unzip : comando di estrazione
- 9082 : nome dell'archivio

Il risultato fu un file di testo "feathers.txt". Visualizzai il suo contenuto con cat.

```
root@kali:/home/kali/Desktop/inventario/_pileoffeathers.jpg-0.extracted# cat feathers.txt
ZncjZr;#7O%FkM2Y2MmRiN2I5MMMyOGI20DEzNzAwMDM5NDYzOWZ9IC0gQ3Jvc3MgdGhLIGNoYXNtIGF0IC9hbWFnaWNiLkZ2VhcHBLYXJzYXR0aGVjaGFzbQ==
```

Apparve, ancora una volta, una stringa codificata "base64" (Parte I - Figura 6).

Utilizzai, come avevo già fatto con il primo messaggio, il comando base64 (concatenandolo con il comando cat . (Figura 22).

```
root@kali:/home# cat feathers.txt | base64 -d ; echo
Flag3{9ad3f62db7b91c28b68137000394639f} - Cross the chasm at /amagicbridgeappearsatthechasm
```

- cat : visualizza il contenuto del file
- txt : nome del file
- |: operatore di concatenazione "pipe" che permette a due processi separati di comunicare fra di loro
- base64 : comando per la decodifica
- - d : opzioni di decodifica
- ; echo : al termine viene stampato a video il risultato dei comandi

Finalmente, ottenni la mia flag numero 3, seguita dall'ormai consueto hash MD5.

Come avevo già fatto in precedenza, "brut-forzai" l'hashcat, per risalire alla password (Parte I - Figura 9).

FLAG3: 9ad3f62db7b91c28b68137000394639f : **345465869** .

## FLAG – 04.

Assieme all'hash c'era anche un altro indizio (Figura 22):

*Attraversa l'abisso: /amagicbridgeappearsatthechasm*

Aveva tutta l'aria di un percorso all'interno dell'URL conosciuto, così mi addentrai nell'oscurità, aggiungendo il percorso */amagicbridgeappearsatthechasm* raggiunsi l'indizio successivo (Figura 23):

