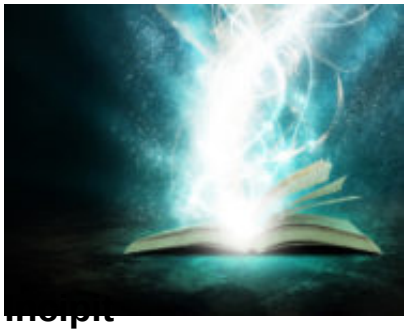
A case history: CTF Necromancer - Parte 2

Author : Vincenzo Digilio

Date : 18 Maggio 2020



Incipit

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 temine 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).

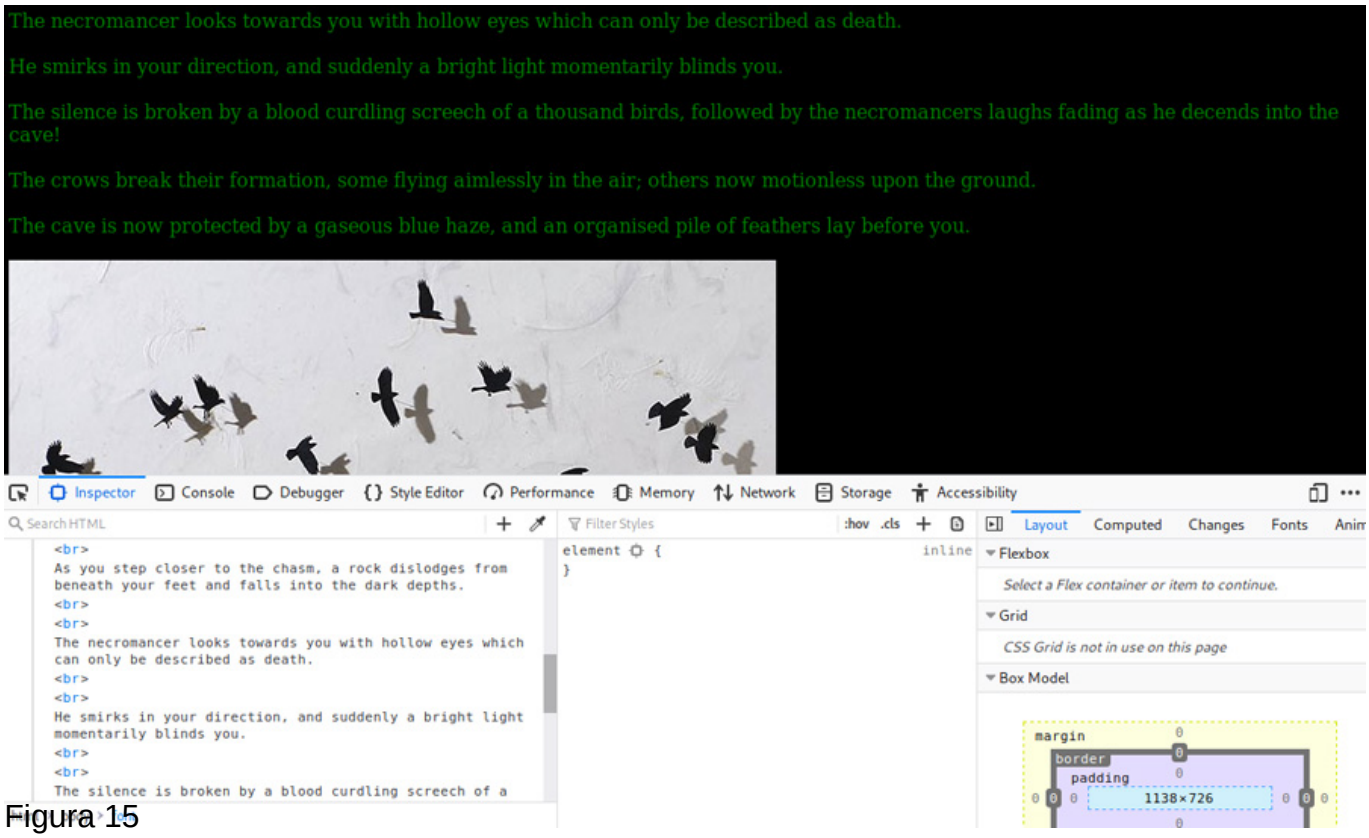


Figura 15

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

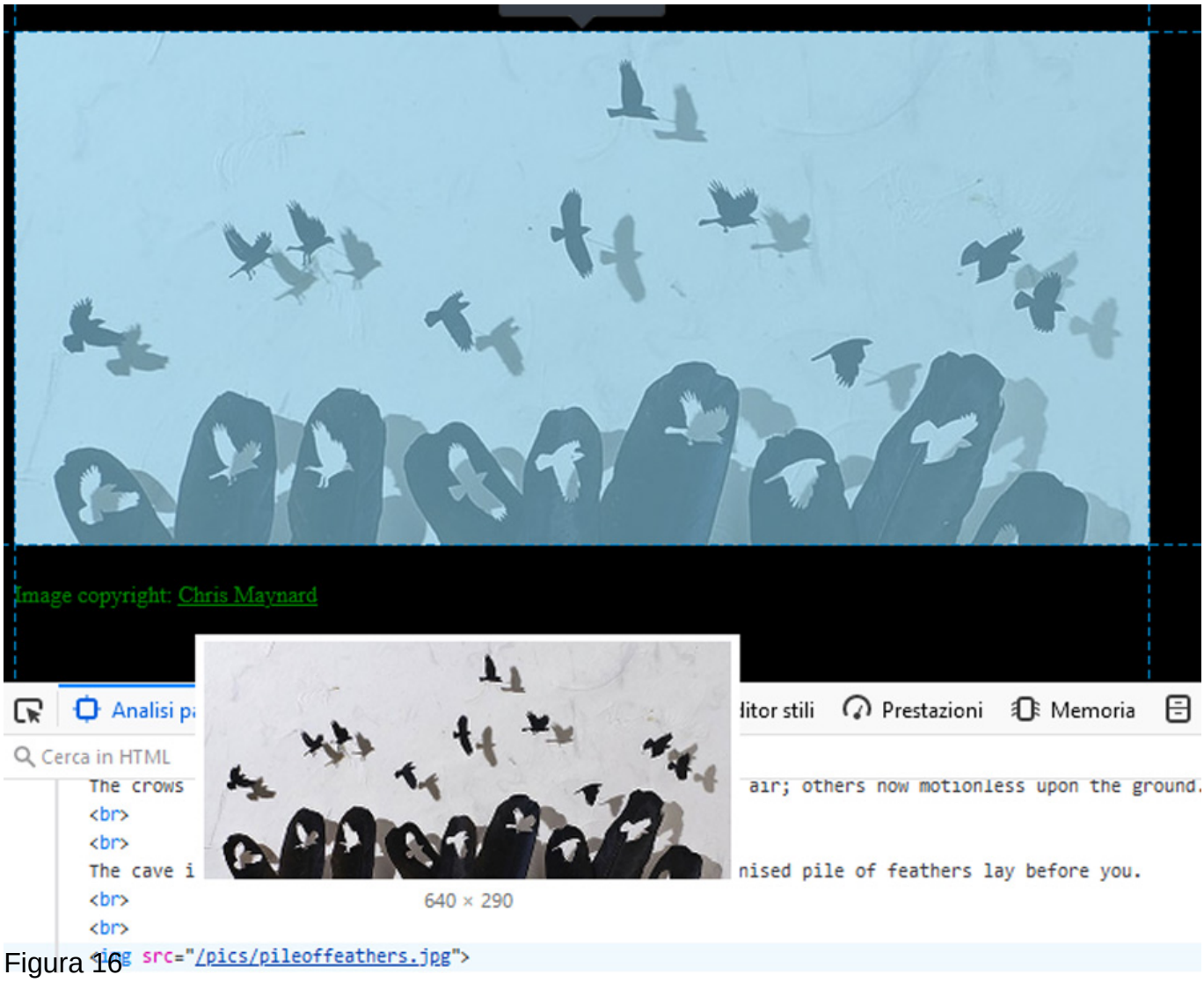


Figura 16

Utilizzai 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).

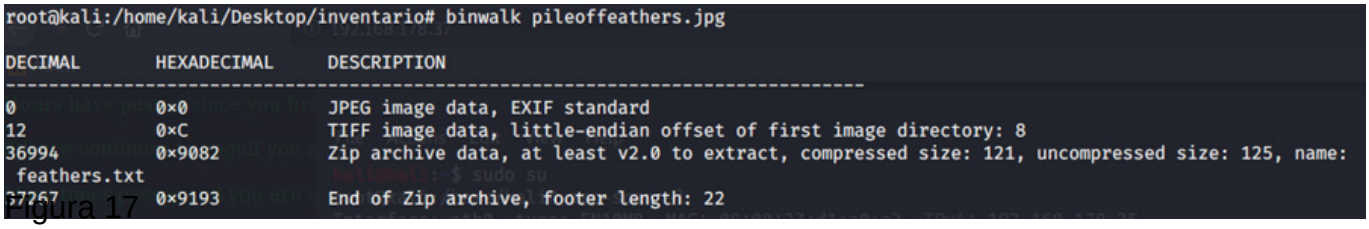


Figura 17

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):

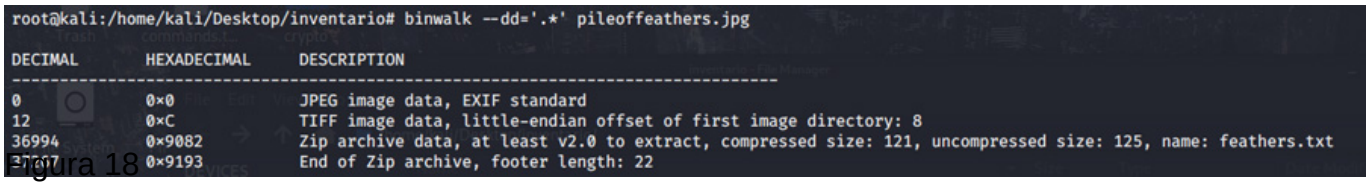


Figura 18

- 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.

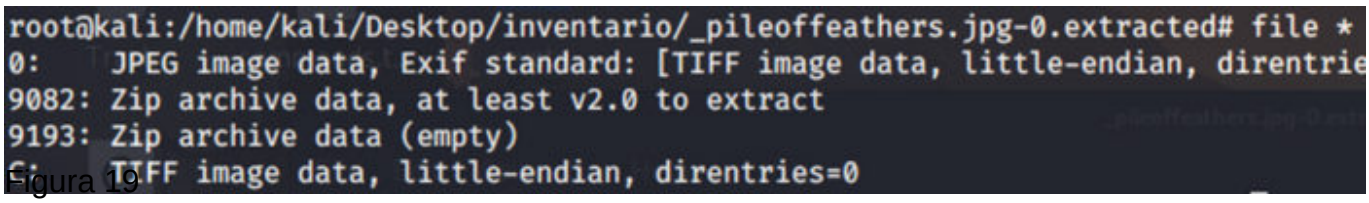


Figura 19

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

Figura 20

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

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

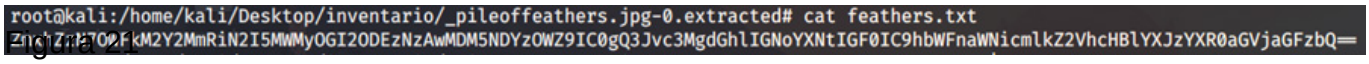


Figura 21

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).

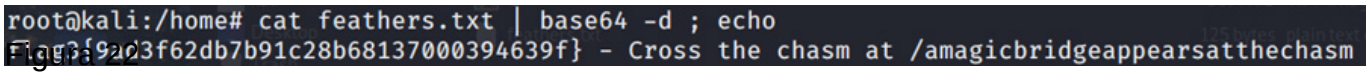


Figura 22

- 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):