

# राश्मिCTF

CHALLENGE NAME: [ Lost Evidence ]

DEV : [ Riya Shah ]

CATEGORY :

[Steganography]

LEVEL: [ Easy ]



2025

### Challenge Description:

A renowned journalist vanished while investigating a secret organization. Days later, an image appeared on their blog before it was taken down. Rumors say it holds the last piece of their research—proof that could shake the world. Can you uncover the hidden truth before it's lost forever?

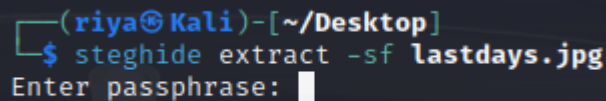
Flag Format: VishwaCTF{} (all lower case)

Attachments: an image(.jpg file)

### Writeup:

- 1) We're given a .jpg file. We analyse it using steghide but are not provided a passphrase:

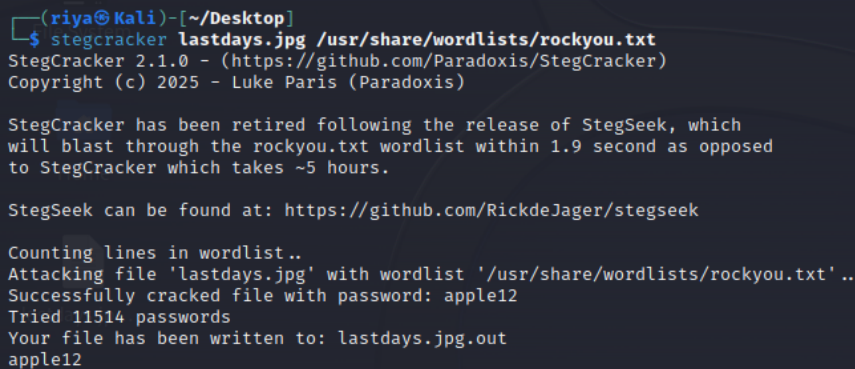
```
steghide extract -sf lastdays.jpg
```



```
(riya@Kali)-[~/Desktop]
$ steghide extract -sf lastdays.jpg
Enter passphrase: █
```

- 2) So perform bruteforce attack on the image using stegcracker with the word list rockyou.txt:

```
stegcracker lastdays.jpg /usr/share/wordlists/rockyou.txt
(password received:apple12)
```



```
(riya@Kali)-[~/Desktop]
$ stegcracker lastdays.jpg /usr/share/wordlists/rockyou.txt
StegCracker 2.1.0 - (https://github.com/Paradoxis/StegCracker)
Copyright (c) 2025 - 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 'lastdays.jpg' with wordlist '/usr/share/wordlists/rockyou.txt'..
Successfully cracked file with password: apple12
Tried 11514 passwords
Your file has been written to: lastdays.jpg.out
apple12
```

- 3) A binary file is received.

4)Now performing:

```
BINARY:0011100000110110001000000011000100110000001101010010000000110001001100
01001101010010000000110001001100000011010000100000001100010011000100111001001
00000001110010011011100100000001101100011011100100000001110000011010000100000
00110111001100000010000000110001001100100011001100100000001110000011001100100
00000110001001100000011010000100000001110010011011100100000001100010011000000
11000000100000001101000011100000100000001100010011000100111001001000000011000
10011000100110101001000000011100100110101001000000011011100110010001000000011
01000011100100100000001100010011000000110000001000000011000100110000001100010
01000000011100100110101001000000011100000110100001000000011000100110001001101
00001000000011000100110001001101110010000000110001001100010011011000100000001
1000100110000001101000010000000110001001100100011010100100000
```

(Binary->Hex)

```
HEX:38 36 20 31 30 35 20 31 31 35 20 31 30 34 20 31 31 39 20 39 37 20 36 37
20 38 34 20 37 30 20 31 32 33 20 38 33 20 31 30 34 20 39 37 20 31 30 30 20 34
38 20 31 31 39 20 31 31 35 20 39 35 20 37 32 20 34 39 20 31 30 30 20 31 30 31
20 39 35 20 38 34 20 31 31 34 20 31 31 37 20 31 31 36 20 31 30 34 20 31 32 35
20
```

(Hex->ASCII)

```
ASCII: 86 105 115 104 119 97 67 84 70 123 83 104 97 100 48 119 115 95 72 49
100 101 95 84 114 117 116 104 125
```

(ASCII->Text)

TEXT:VishwaCTF{Shad0ws\_H1de\_Truth}

**Flag: VishwaCTF{Shad0ws\_H1de\_Truth}**