

LinkedIn: [Kelvin Kimotho](#)

CYBER TALENTS

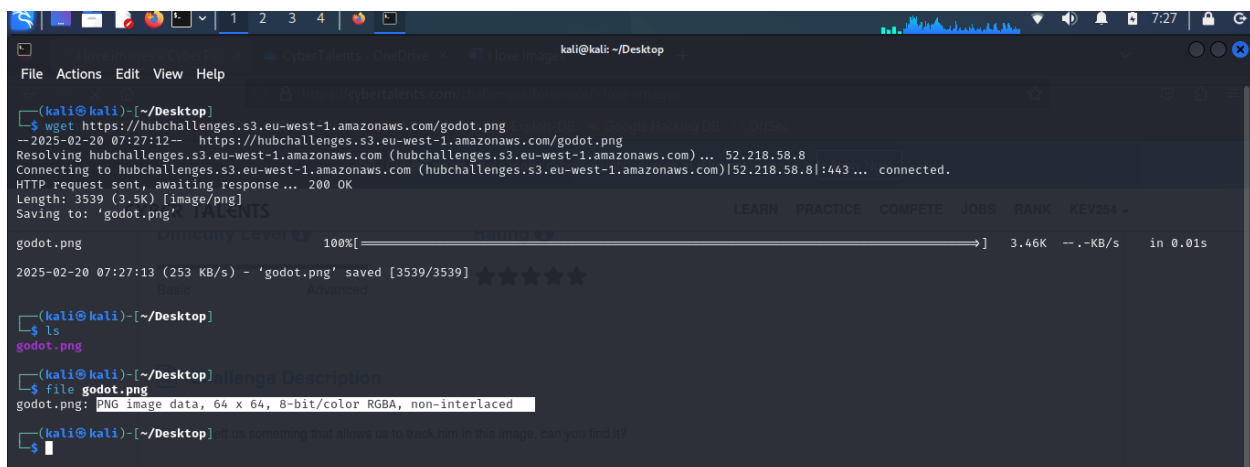
Challenge Name: I love images

Challenge Description

A hacker left us something that allows us to track him in this image, can you find it?

Solution

I first downloaded the image file into my machine using **wget** a command line tool used to download files from the internet.



```
(kali@kali)-[~/Desktop]
└─$ wget https://hubchallenges.s3.eu-west-1.amazonaws.com/godot.png
--2025-02-20 07:27:12-- https://hubchallenges.s3.eu-west-1.amazonaws.com/godot.png
Resolving hubchallenges.s3.eu-west-1.amazonaws.com (hubchallenges.s3.eu-west-1.amazonaws.com) ... 52.218.58.8
Connecting to hubchallenges.s3.eu-west-1.amazonaws.com (hubchallenges.s3.eu-west-1.amazonaws.com)|52.218.58.8|:443 ... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3539 (3.5K) [image/png]
Saving to: 'godot.png'
100%[====================================================>] 3.46K --KB/s in 0.01s

2025-02-20 07:27:13 (253 KB/s) - 'godot.png' saved [3539/3539]

(kali@kali)-[~/Desktop]
└─$ ls
godot.png

(kali@kali)-[~/Desktop]
└─$ file godot.png
godot.png: PNG image data, 64 x 64, 8-bit/color RGBA, non-interlaced

(kali@kali)-[~/Desktop]
└─$
```

I then went ahead and used **exiftool** a command line tool that enables us examine files meta data but i found nothing interesting.

```
(kali@kali)-[~/Desktop]
└─$ xiftool godot.png
ExifTool Version Number      : 12.76
File Name                    : godot.png
Directory                    : .
File Size                    : 3.5 kB
File Modification Date/Time   : 2024:11:27 12:57:26+00:00
File Access Date/Time        : 2025:02:20 07:27:13+00:00
File Inode Change Date/Time   : 2025:02:20 07:27:13+00:00
File Permissions              : -rw-rw-r--
File Type                    : PNG
File Type Extension          : png
MIME Type                    : image/png
Image Width                  : 64
Image Height                 : 64
Bit Depth                    : 8
Color Type                   : RGB with Alpha
Compression                  : Deflate/Inflate
Filter                       : Adaptive
Interlace                    : Noninterlaced
Warning                      : [minor] Trailer data after PNG IEND chunk
Image Size                   : 64x64
Megapixels                   : 0.004
```

I then went ahead looking for possible strings embedded into the image, this was possible by using **strings** a command line tool that *looks for printable strings in a file*. I came across some suspicious string 'IZGECR33JZXXIX2PNZWHSX2CMFZWKNRUPU===== ' which appeared to be a base64 encoding.

```
(kali@kali)-[~/Desktop]
└─$ strings godot.png
IHDR
qIDATx
{pTU
\vvgtwj
QA!~
%EY}
4:$YF
B I2*

vL.I
VMqa.%
pDuF(
op+P
q}'ZA0
O-T6
IEND
IZGECR33JZXXIX2PNZWHSX2CMFZWKNRUPU=====
```

I went ahead and tried decoding it using the **base64 decoder** tool on my machine but it turned out that the encoding was not a base64 encoding.

```
(kali@kali)-[~/Desktop]
└─$ echo IZGECR33JZXXIX2PNZWHSX2CMFZWKNRUPU===== | base64 -d
!.. *%*!}*5*I}*0VV(*T=base64: invalid input
```

I tried doing some research regarding encodings. Went ahead and made a prompt asking ai what encoding was the string and it identified it as a **base32** encoding.

The string "IZGECR33JZXXIX2PNZWHSX2CMFZWKNRUPU=====" is a base32 encoded text. Base32 encoding is often used for encoding binary data in a way that is more human-readable and can be easily transmitted in text formats.

👍 🗑️ ↺ 📄 📝 >> 📌 📁 🕒 +

Enable Notifications

Message Blackbox or @mention agent



🌐 Web Search 📖 Deep Research ⚙️ Models ⚡ Beast Mode 🖼️ Image 📤 Upload ⚙️ Customize + Multi-Panel

I then used the base32 decoder on my machine and captured flag was "

FLAG{Not_Only_Base64}".

```
FLAG{Not_Only_Base64}
(kali@kali)-[~/Desktop]
└─$
(kali@kali)-[~/Desktop]
└─$
```