Today's Challenge: Forensics Meets Cryptography

I tackled a fascinating forensics challenge with a cryptographic twist today. The challenge kicked off with a hint from the author:

"I don't like scrolling down to read the code of my website, so I've squished it. As a bonus, my pages load faster!"

Along with the hint, a JPG image was provided. At first glance, the image appeared unremarkable, but as is often the case with Capture The Flag (CTF) challenges, the surface doesn't tell the whole story.

Step 1: Initial Analysis

To begin, I uploaded the image to **Aperi Solve**, an online tool designed for in-depth image analysis. Aperi Solve provides features like viewing embedded strings, detecting hidden files, and utilizing tools like *steghide* to uncover concealed data.

The analysis revealed that **steghide** had successfully extracted hidden data and saved it to a file named flag. Upon downloading the file, I noticed it lacked a recognizable extension, suggesting it might be raw or default data.

Step 2: Identifying the File Type

A quick check using an online file type tool identified the file as an ASCII file. Based on this, I renamed it with a .txt extension. Opening the file revealed the following hint:

"The flag is not here; maybe think in simpler terms. Data that explains data."

The hint suggested looking for something more straightforward.

Step 3: Inspecting Metadata

I returned to Aperi Solve and examined the image's **EXIF metadata**. In the attributionURL field, I found a suspicious string:

cGljb0NURntNRTc0RDQ3QV9ISUREM05fZDhjMzgxZmR9Cg==

This string had all the hallmarks of **Base64 encoding**.

Step 4: Decoding the Flag

I copied the string and turned to **CyberChef**, a versatile tool for cryptographic operations. Using the *Base64 decode* function, I processed the string, which revealed the flag:

picoCTF{ME74D47A_HIDD3N_d8c381fd}

Key Takeaways

This challenge was an excellent reminder that data can be hidden in the most unexpected places. It emphasized the importance of exploring every aspect of a file, from its embedded strings to its metadata, when searching for hidden information.

Thank you for reading today's writeup! See you tomorrow with another challenge!