Transformation

In this challenge, there was a file given that contained unusual Unicode characters and the encoding logic that was used to create those Unicode characters. The encoding logic worked by shifting the first character 8 bits to the left and then adding the second character. The result of this expression was then converted into a single Unicode character. This encoded each pair of ASCII characters into one Unicode character. In order to decode this file, each Unicode character was converted into its numerical value and then separated into high and low bytes using bitwise operations. These were then converted into characters to reveal the original flag, picoCTF{UTF16_is_weird_inst34d_of_8}.This CTF challenge that I worked on involved cat.jpg, where the hint was "look at the details of the file." After closer inspection, the image looked legitimate, but after analyzing its information through exiftool and looking through its raw data through xxd, it indeed contained a Base64 string: "cGljb0NURnt0aGVfbTN0YWRhdGFfMXNfbW9kaWZlZWR9." Which, after decoding, returned "picoCTF{the_m3tadata_1s_modified}." The lesson to take away from this challenge is how flags can hide themselves away from the raw information contained within a document, and how it is relatively straightforward after employing the aid of commands such as 'file', 'exiftool'.