1.b_challenge4.docx:

strings b_challenge4.docx and scroll up to find
__flag{h0wz_the_joke_hahahha!!}

2.E_challenge4.jpg:

>exiftool E_challenge4.jpg

#in the comment section there's "not so human readable form"
Njk2ZTYzNzQ2NjdiNzkzMDc1NWY2NzMwNzQ1ZjM3NjgzMzNDY3N2Q=
#then if you see this code carefully it is in the base64 form
#then go to any conversion site via google convert it into every form possible
then in the conversion to ascii we get -
'696e6374667b7930755f6730745f3768335f666c34677d'
it looks like its in hexadecimal form
convert it once more then we get--
__inctf{y0u_g0t_7h3_fl4g}

3.FS_challenge4.png:

>ghex FS_challenge4.png

in the new window in the first line itself there is a problem with the magic number:

so if we correct it then save it the png will open with the flag:

__inctf{7h4nk5_for_h3lping_m3}

4.s_challenge4.png:

>strings s_challenge4.png

#in the output of the following.
there is something different with second last line.
if we see it clearly then we understand it is of hexadecimal form -
5a6d78685a33746f4e474e724d334a7a587a52794d31387a646a4e796558646f4d33497a66513d3d
if we see the ascii text... it is - ZmxhZ3toNGNrM3JzXzRyM18zdjNyeXdoM3IzfQ==
it looks like base64 format then if we convert it into ascii text format then

___flag{h4ck3rs_4r3_3v3rywh3r3}

5.SH_challenge4.jpg

>steghide extract -sf SH_challenge4.jpg

then it asks the passphrase that we don't even know!!

SO ,to get the passphrase

>strings SH_challenge4.png

#in the output of this the last line looks different copy it.

then do --
>md5sum SH_challenge4.png

then see that the output and the line that you copied is of same length. -
('723fa61abce2c64e60f5f3a4c1426a15')

if we were to use the md5 algorithm,we get the output --- ('WEAREFREE')

try this as the passphrase

the if we type the passphrase on: >steghide extract -sf SH_challenge4.jpg -xf 1.txt
then the extracted info gets pasted onto 1.txt
if we open 1.txt then we get the flag:

__inctf{H4pPy_Ind3p3nD3nC3_D4Y}

6.SS_challenge4.jpg

>Now we should open file via stegsolve.jar

if we navigate through every type of plane and keep saving files in folder

and after that keep using zbarimg on every file available

after that we find the flag in one of the files in one of the files

>zbarimg red0.bmp

which gives the flag.

__VolgaCTF{5t3g0_m4tr3shk4_in_4cti0n}

7.Z_challenge4.png

>zbarimg Z_challenge4.jpg

if we do this we get the flag.

__flag{g00d_j0b_g33ks!!}