**Task** - At first glance, this image seems ordinary — but something doesn't add up. A closer look might reveal a hidden secret, carefully tucked away from plain sight.

Your task is to investigate the image, uncover any hidden data, and retrieve a concealed message. But the challenge doesn't stop there — the message is encrypted, and you'll need to decrypt it to reveal the final flag.

Attachment - ACE1234.jpg

ANSWER:-

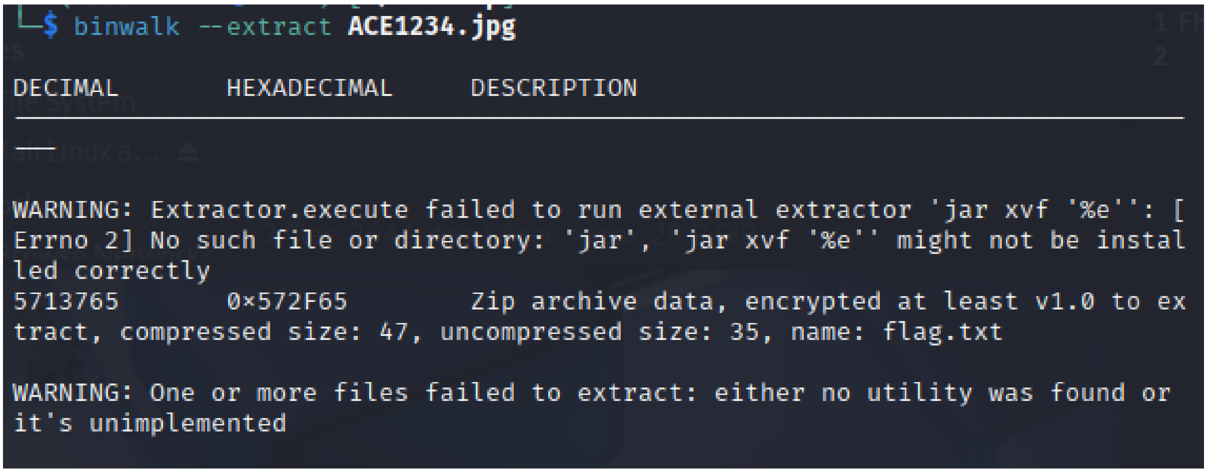
## Methodology

The solution was achieved in two main phases: first, uncovering the hidden data through **steganography**, and second, decrypting the concealed message and its prefix through **cryptography**.

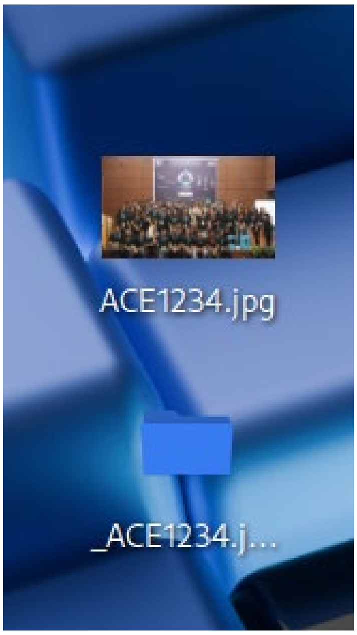
### Phase 1: Steganography - Finding the Hidden File

This phase of the solution remains the same, as it was the correct path to finding the encrypted data.

1. **File Analysis:** The `binwalk` tool was used to scan the image file `ACE1234.jpg` for embedded data.
   - **Command:** `binwalk --extract ACE1234.jpg`
   - **Result:** The command successfully identified and extracted a password-protected zip archive named **`572F65.zip`** from within the image.



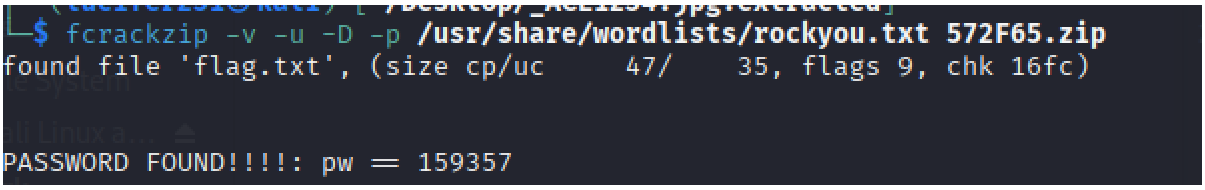
```
└─$ binwalk --extract ACE1234.jpg

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
──

WARNING: Extractor.execute failed to run external extractor 'jar xvf '%e'': [
Errno 2] No such file or directory: 'jar', 'jar xvf '%e'' might not be instal
led correctly
5713765       0x572F65        Zip archive data, encrypted at least v1.0 to ex
tract, compressed size: 47, uncompressed size: 35, name: flag.txt

WARNING: One or more files failed to extract: either no utility was found or
it's unimplemented
```



## Phase 2: Cryptography - Decryption

This phase involved finding the password for the zip archive and then decrypting the message contained within it.

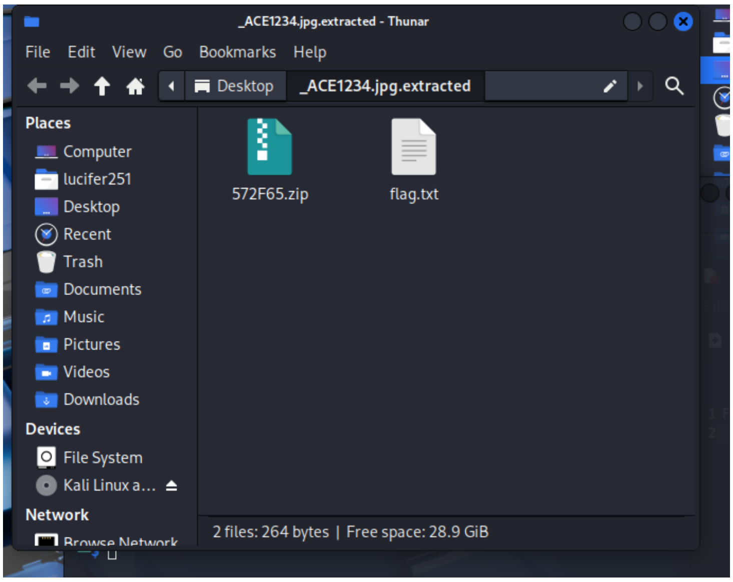
1. **Password Discovery:**
   - The archive `572F65.zip` was encrypted, requiring a password. A dictionary attack was performed using the **`fcrackzip`** tool pointed at the comprehensive `rockyou.txt` password list.
   - **Command:** `fcrackzip -v -u -D -p /usr/share/wordlists/rockyou.txt 572F65.zip`
   - **Result:** The command successfully cracked the archive's password, revealing it to be **`159357`**.



```
└─$ fcrackzip -v -u -D -p /usr/share/wordlists/rockyou.txt 572F65.zip
found file 'flag.txt', (size cp/uc     47/    35, flags 9, chk 16fc)


PASSWORD FOUND!!!!: pw == 159357
```

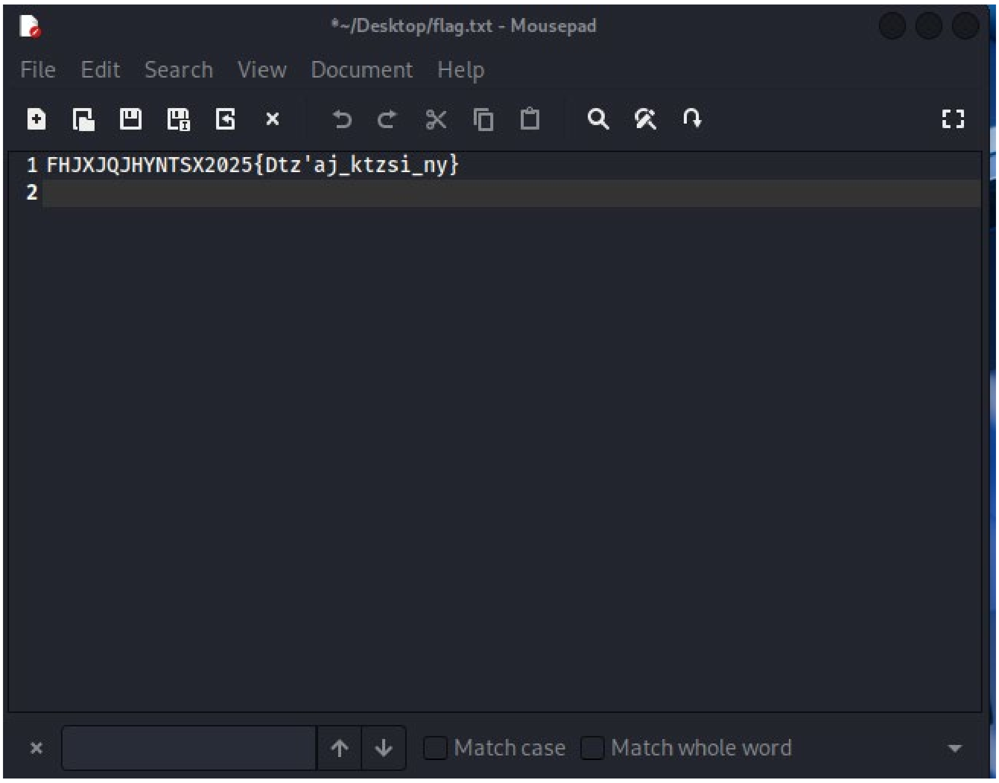
2. **File Extraction:**

- Using the found password (`159357`), the contents of `572F65.zip` were extracted.
- **Result:** This produced a single file named `flag.txt`.

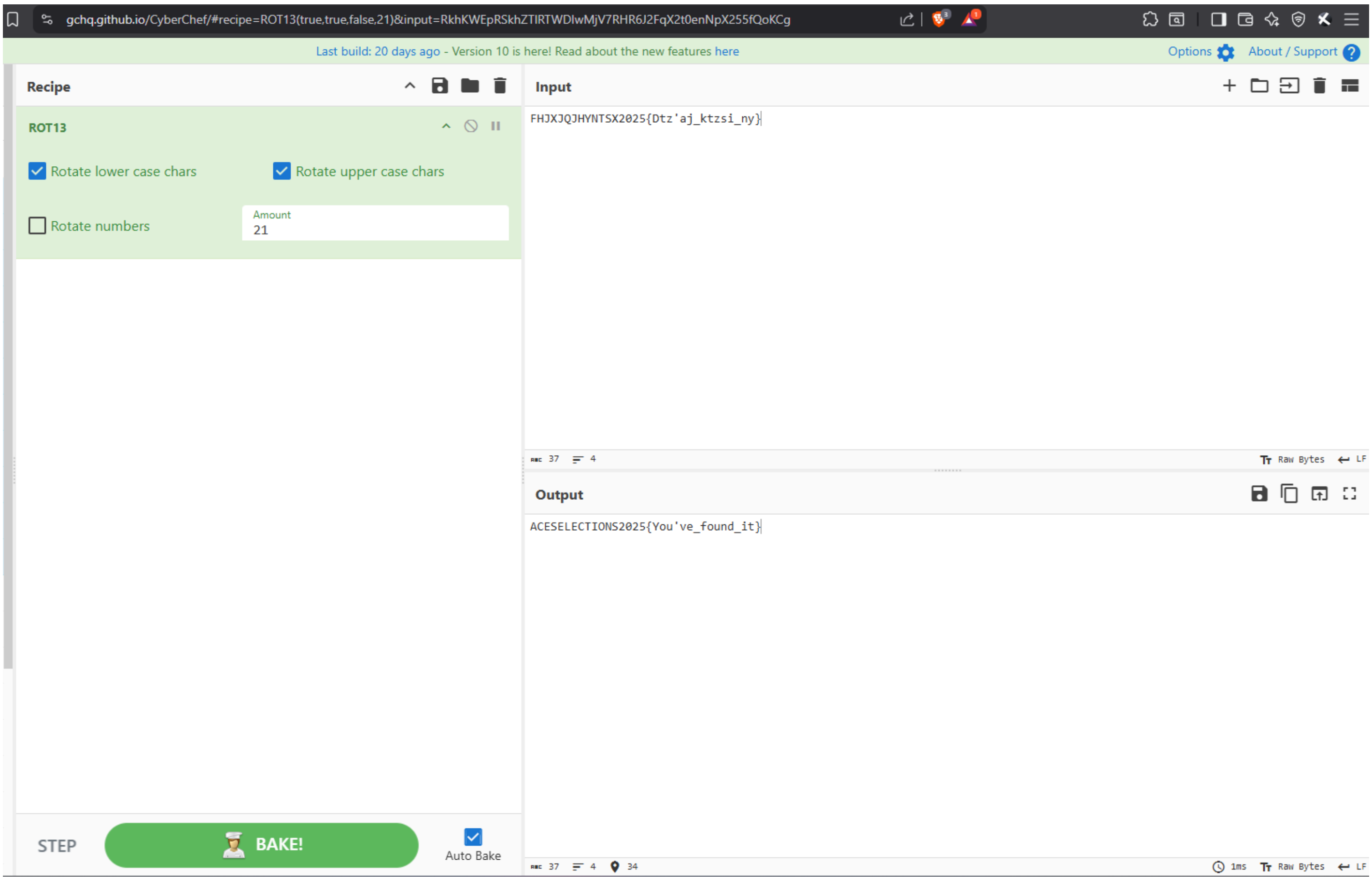


3. **Final Flag Decryption:**

- The `flag.txt` file contained the string `FHJXJQJHYNTSX2025{Dtz'aj_ktzsi_ny}`. This entire string, including the prefix, was encrypted.
- The encryption was identified as a **Caesar cipher**. Using **CyberChef**, the ROT13 operation was selected.
- By setting the shift **Amount** to **21**, the entire string was correctly decrypted.
- **Result:** The plaintext was revealed to be `ACESELECTRONS2025{you've_found_it}`.


*~/Desktop/flag.txt - Mousepad
File Edit Search View Document Help
Match case
Match whole word

```
FHJXJQJHYNTSX2025{Dtz'aj_ktzsi_ny}
```


Last build: 20 days ago - Version 10 is here! Read about the new features here
Options
About / Support
Recipe
Input
Output
STEP
BAKE!
Auto Bake

gchq.github.io/CyberChef/#recipe=ROT13(true,true,false,21)&input=RkhKWEpRSkhZTlRTWDIwMjV7RHR6J2FqX2t0enNpX255fQoKCg

ROT13

☑ Rotate lower case chars ☑ Rotate upper case chars

□ Rotate numbers

Amount
21

```
FHJXJQJHYNTSX2025{Dtz'aj_ktzsi_ny}
```

```
ACESELECTIONS2025{You've_found_it}
```