

# OSINT CHALLENGE 2

---

## Challenge Write-Up: *Whispers on the Express*

### Challenge Description Recap

Every great journey leaves traces beyond the photograph.

Some are etched in stories, others hidden in conversations left behind.

This moment was shared publicly by **wanderingwizard** where people share opinions,

and a quiet signal was added where people usually speak — not where images live.

Listen closely to what was said, not just what was shown.

The message speaks in pulses, and what you hear is only the beginning.

Decode the layers to uncover the truth.

**Flag format:** EXC{...}

---

### Understanding the Clues (Plain English)

Let's break this down slowly:

- “**shared publicly by wanderingwizard**”
  - Refers to a public post by a user named **wanderingwizard**
- “**where people share opinions**”
  - This points to **Twitter** (now X)
- “**not where images live**”
  - The clue is *not* in the image itself
  - It is in the **comments / replies**
- “**speaks in pulses**”

- Strong hint for **Morse code**
- “**what you hear is only the beginning**”
- Implies **multiple layers of decoding**

So we already know:

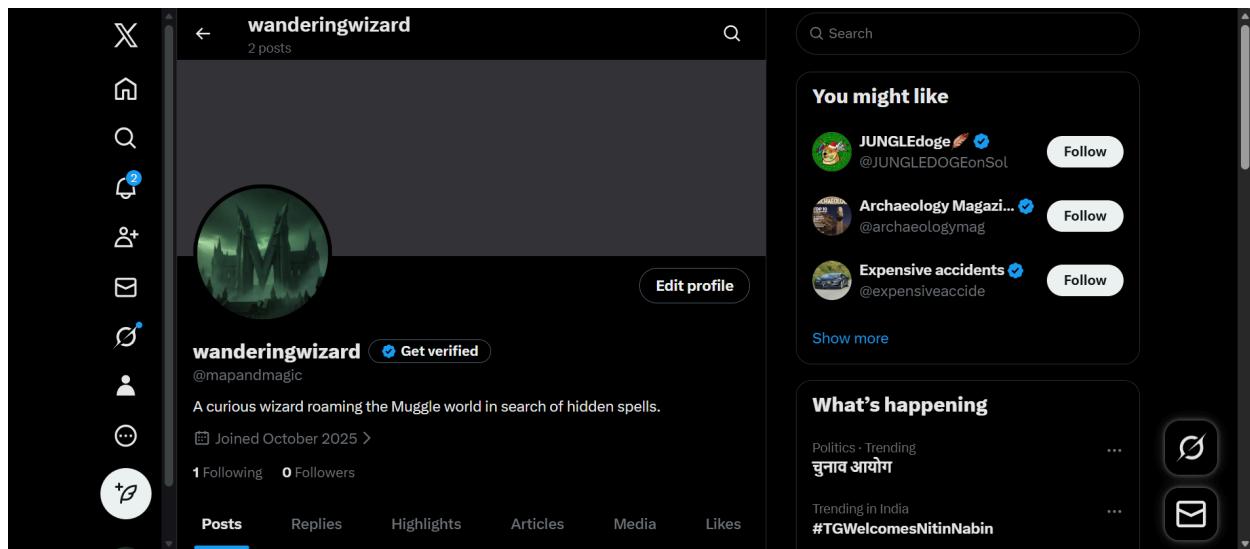
The flag is hidden in the **comments of a Twitter post**, and it is **encrypted in multiple layers**.

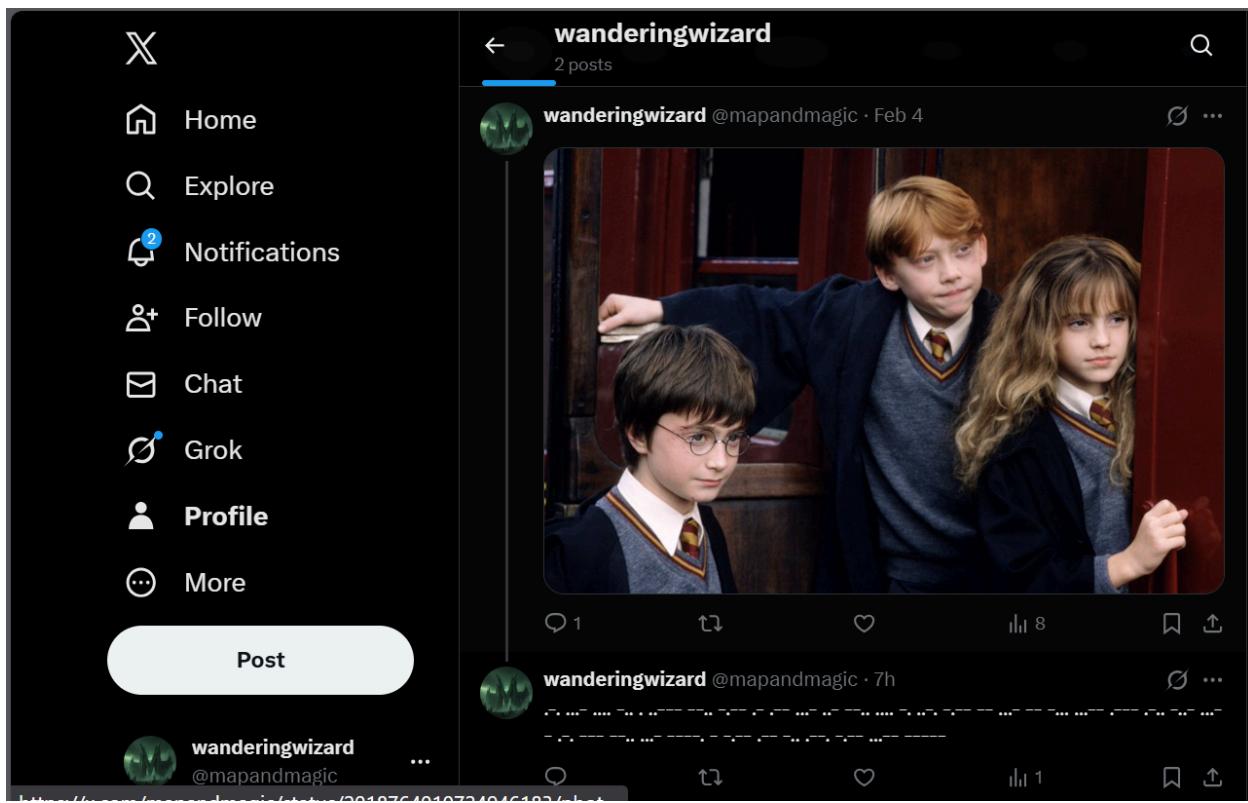
## Step 1: Find the Relevant Twitter Post

1. Go to Twitter (X)
2. Search for the user: `wanderingwizard`
3. Look through their public posts related to the challenge theme
4. Open the post and **carefully read the comments/replies**

You will find a comment that looks unusual — consisting of dots, dashes, or strange symbols.

This is **not normal text**, which confirms it is encrypted.





## Step 2: Identify the Encryption Layers

From the challenge and confirmation:

1. **First layer:** Base64
2. **Second layer:** Morse Code

Important:

The order matters. You must decode **in the correct sequence**.

## Step 3: Decode the Morse Code

Take the encrypted comment and:

1. Copy the dots (.) and dashes (-)
2. Use any Morse decoder (online or offline)
3. Decode the Morse code into readable text

After decoding Morse, the output will **still look scrambled**.

That's expected — because Morse is **not the final layer**.

The image contains two side-by-side screenshots of web-based Morse code translation tools.

**Left Screenshot:** A search interface for 'dCode' with a search bar containing 'caesar'. Below it, a results section shows the Morse code representation of the word 'caesar':

```
!!  
(- FH  
(-- QOT  
(.- VSNNUIBNAT◆TATENTFAEBT  
(.- OJAAGMJANE◆ENETAEQNTJE  
RVHDE2ZYAWUZHNFYVMB3JLX3ROZV9TYWDPY3  
0  
(.- RVHDE2ZYAWUZHNFYVMB3JLX3ROZV9TYWDPY3  
0  
(.- KÖCHWT7ÜLDÖGUCHAQLIÖIJ8BYP8KSÜÖ4ELDWX  
L85  
# #7
```

**Right Screenshot:** A 'MORSE CODE TRANSLATOR' tool. It has a text input field containing Morse code and a 'TRANSLATE AUTOMATICALLY' button. The interface includes sections for 'MORSECODE CHARACTERS' (radio buttons for 'USE PERIOD . FOR SHORT AND DASH - FOR LONG' and 'USE THE FOLLOWING CHARACTERS FOR SHORT / DOT di and CHARACTER(S) FOR LONG / DASH dah'), 'MORSE SPACE MANAGEMENT' (radio buttons for 'THE MESSAGE HAS SPACES BETWEEN EACH MORSE CODE' and 'THE MESSAGE USES THIS SEPARATOR: |'), and a 'Summary' sidebar on the right with links related to Morse code.

## Step 4: Decode Base92

Now take the output from the Morse decoding step and:

1. Identify it as **Base64-encoded text**
2. Use a Base64 decoder
3. Decode the text

After decoding Base64, you finally get a **clean, readable sentence**.

The screenshot shows two main sections of the dCode website. On the left, a search bar is at the top with the placeholder "Search for a tool" and a sub-section for "SEARCH A TOOL ON DCODE" with a dropdown menu showing "e.g. type 'sudoku'". Below this is a "Results" section containing several lines of text, each starting with "EXC{" followed by variations of "friends\_before\_the\_magic". On the right, there is a "BASE64 CODING" page with a "BASE64 DECODER" form. The "CIPHERTEXT" field contains the string "RVHDE2ZYAWU.Y30". Below it, the "MODE" section has three options: "BASE64 (STANDARD RFC 4648)" (radio button), "BRUTEFORCE: TRY ALL BASE64 VARIANT (SEE FAQ)" (radio button), and "NO CASING: UPPER-LOWERCASE IS WRONG/LOST (BRUTEFORCE MAX 50 CHARS)" (radio button, selected). The "RESULTS FORMAT" section has four options: "STRING OF PRINTABLE CHARACTERS (ASCII/UNICODE)" (radio button, selected), "HEXADECIMAL 00-FF" (radio button), "DECIMAL 0-127-255" (radio button), and "000-177-377" (radio button). To the right of the decoder form is a "Summary" sidebar with links related to Base64 encoding and decoding.

## Step 5: Extract the Flag

The fully decoded text reveals the flag content:

```
EXC{friends_before_the_magic}
```

This matches the correct solution

## Final Answer

Flag:

```
EXC{friends_before_the_magic}
```

## Key Takeaways for Beginners

- OSINT clues often hide **outside the obvious content**
- Always check:
  - comments
  - replies

- captions
  - “Pulses”, “signals”, and “listening” often hint at **Morse**
  - Multi-layer encryption is common — **don’t stop after one decode**
  - Order of decoding **matters**
-