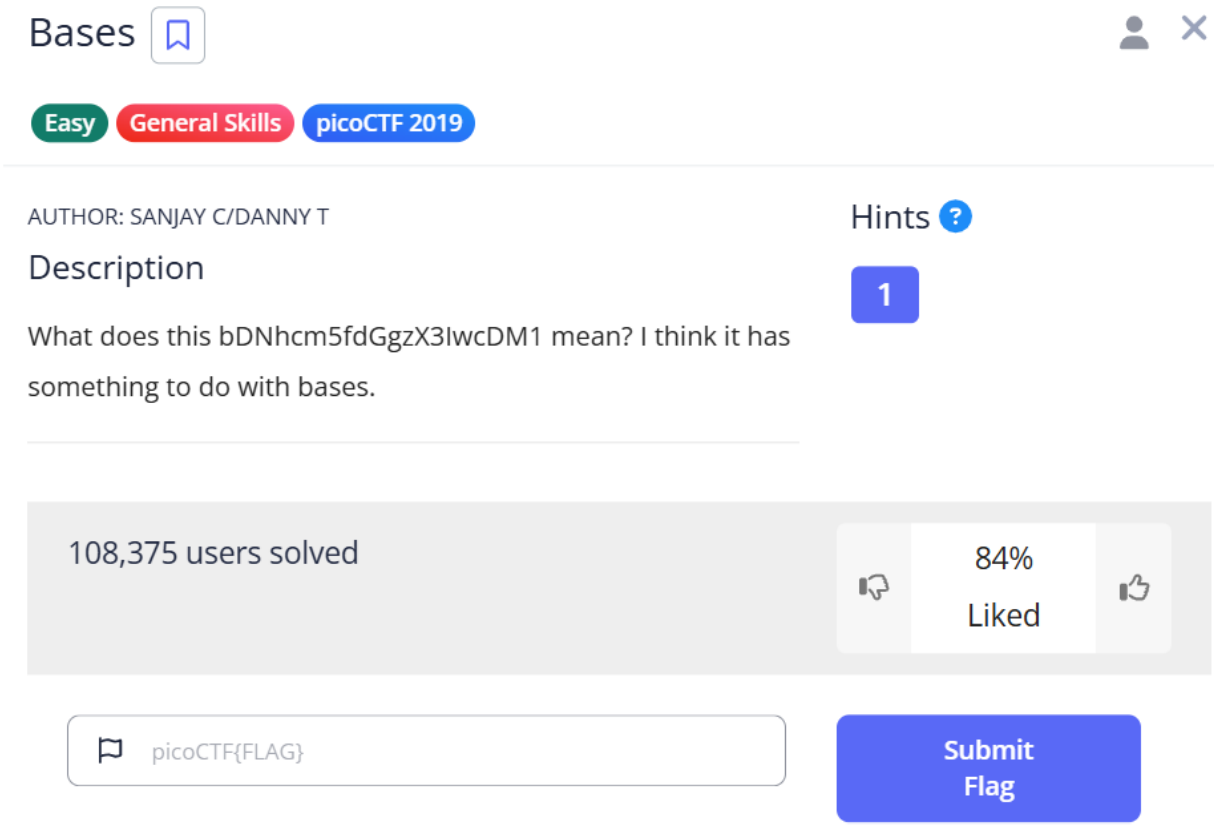
Bases



Author: The Analyst: Hyposelenia

Challenge: Bases

Category: General Skills

Date: 01/13/26

I. Objective

The objective of this challenge was to identify and decode an encoded string using the correct **base encoding** method in order to reveal the hidden flag.

II. Background

In cybersecurity challenges, information is often hidden using **encoding**.
Encoding changes readable text into another form so it looks confusing.

One common encoding method is **Base64**, which is often used to safely store or transmit data. This challenge required recognizing the encoding type and decoding it correctly.

III. Tool Used

- **Oracle VirtualBox (Kali Linux)** – Used as the Linux environment
- **Linux Terminal** – Used to execute decoding commands
- **base64** – Command-line tool used to decode Base64 strings

IV. Methodology

1. The challenge provided the encoded string:

   bDNhcm5fdGgzX3IwcDM1

2. The description mentioned **"bases"**, suggesting the string is encoded using a base system.
3. The string contained:

Uppercase letters

Lowercase letters

Numbers

4. This is a strong indicator of **Base64**, since Base64 uses:

   A–Z

   a–z

   0–9

   + and / (sometimes omitted)

5. The string length is also a multiple of **4**, which is another clue for Base64 encoding.
6. The string was decoded using the following Linux command:

   echo bDNhcm5fdGgzX3IwcDM1 | base64 -d

7. The output of the command revealed the decoded text:

   l3arn_th3_r0p35

8. The decoded text was placed inside the required flag format.

V. Result

picoCTF{l3arn_th3_r0p35}

VI. Explanation

To understand this challenge, we must first understand **what a "base" is**.
A **base** is simply a set of symbols used to represent data.

- **Base10** → uses numbers 0–9 (what humans normally use)
- **Base2 (binary)** → uses only 0 and 1
- **Base16 (hex)** → uses 0–9 and A–F
- **Base64** → uses:
  - Uppercase letters A–Z
  - Lowercase letters a–z
  - Numbers 0–9
  - Sometimes + and /

The purpose of base encodings is **not to hide information forever**, but to:

- Safely store data
- Send data across systems
- Represent binary data in readable text

Why this is NOT Base16 (Hex)
Hexadecimal (**Base16**) only allows:
0 1 2 3 4 5 6 7 8 9 A B C D E F
However, the given string:
bDNhcm5fdGgzX3IwcDM1
contains:

- Lowercase letters beyond f (like h, r, m)
- Uppercase letters
- Numbers

❌ Hex **cannot** contain those characters
✅ Therefore, this is **not Base16**

Why this is NOT Base32

Base32 only allows:
A–Z and 2–7
The string includes:

- Lowercase letters
- Numbers like 0 and 1

❌ Base32 does **not** allow lowercase letters or 0/1
✅ Therefore, this is **not Base32**

VII. Conclusion

This challenge taught the importance of recognizing common encodings in cybersecurity.

By identifying the structure of the encoded string and using the appropriate decoding tool, the hidden flag was easily recovered.

Understanding Base64 is an essential skill for beginners in CTF challenges, as it frequently appears in real-world scenarios involving data transfer and storage.

— The Analyst: Hyposelenia