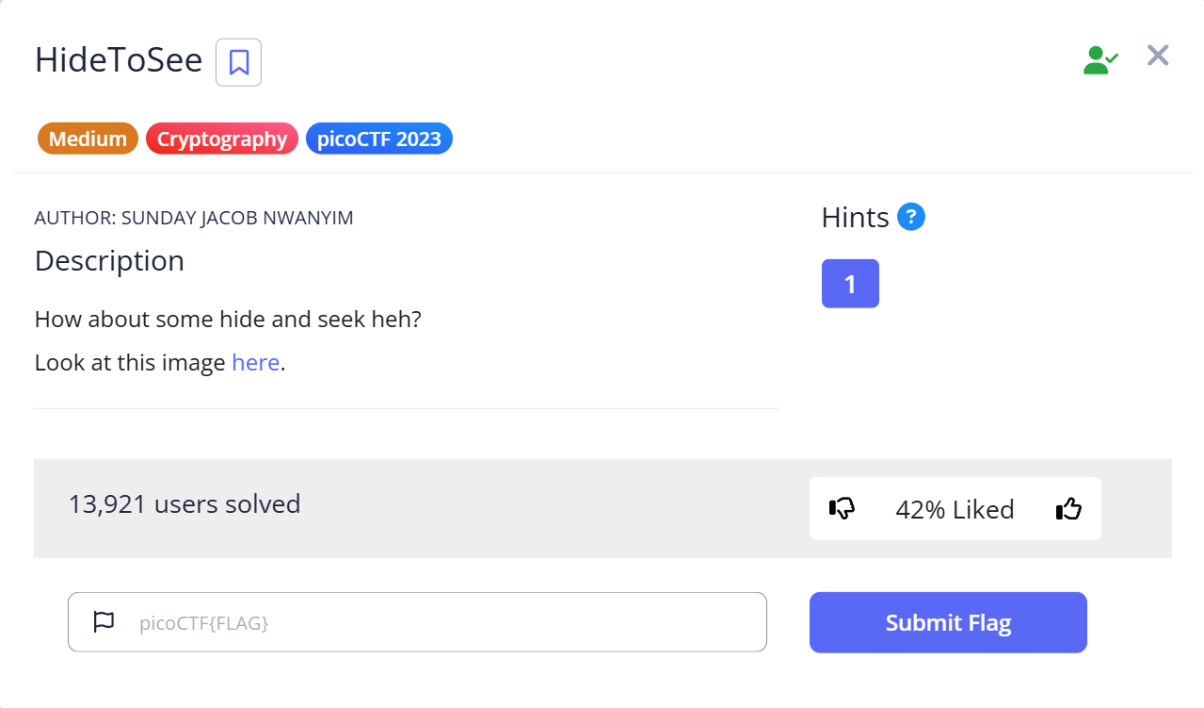
HIDETOSEE

picoCTF 2023
Category: Cryptography (Medium)



CHALLENGE INFORMATION

Challenge Name : HideToSee
Category : Cryptography
Difficulty : Medium
Event : picoCTF 2023
Author : Sunday Jacob Nwanyim
Total Solves : 13,921 users

DESCRIPTION

“How about some hide and seek heh?

Look at this image here.”

File Provided:
atbash.jpg

SOLUTION OVERVIEW

1. Extract hidden data from the provided image using steganography techniques.
2. Decrypt the extracted ciphertext using the Atbash cipher.
3. Obtain the final flag.

DETAILED WALKTHROUGH

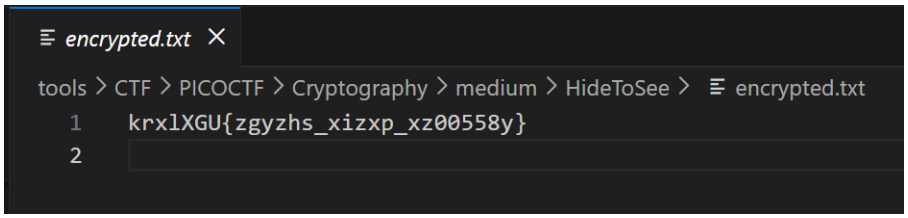
Step 1: Steganography Analysis

The challenge provides an image file named atbash.jpg.
The filename itself serves as a strong hint that the Atbash cipher is involved in this challenge.

To extract hidden data from the image, the steghide tool is used.
When prompted for a passphrase, an empty passphrase is applied.

```
┌──(CYBER)(tel㉿ELELEL)-[~/tools/ctf/picoctf/Cryptography/medium/HideToSee]
└─$ steghide extract -sf atbash.jpg
Enter passphrase:
wrote extracted data to "encrypted.txt".
```

As a result, a file named encrypted.txt is successfully extracted, containing the following ciphertext:



krxlXGU{zgyzhs_xizxp_xz00558y}

Step 2: Understanding the Atbash Cipher

Atbash is a classical substitution cipher in which each letter of the alphabet is replaced by its corresponding reversed letter.

A is substituted with Z
B is substituted with Y
C is substituted with X
and so on.

This cipher is symmetric, meaning that the same transformation is used for both encryption and decryption.

Step 3: Decryption Process

To automate the decryption process, a Python script is used to apply the Atbash transformation to each character in the ciphertext.

Encrypted Text:
krxlXGU{zgyzhs_xizxp_xz00558y}

Python Script Used:

```
def atbash(text):
result = []
for char in text:
if 'a' <= char <= 'z':
result.append(chr(ord('z') - (ord(char) - ord('a'))))
```

```
elif 'A' <= char <= 'Z':
result.append(chr(ord('Z') - (ord(char) - ord('A'))))
else:
result.append(char)
return ''.join(result)

if name == "main":
encrypted = "krxlXGU{zgyzhs_xizxp_xz00558y}"
decrypted = atbash(encrypted)

print("Encrypted:", encrypted)

print("Decrypted:", decrypted)
```

Decrypted Result:

```
┌──(CYBER)(tel㉿ELELEL)-[~/tools/ctf/picoctf/Cryptography/medium/HideToSee]
└─$ python3 solve_atbash.py
Encrypted: krxlXGU{zgyzhs_xizxp_xz00558y}
Decrypted: picoCTF{atbash_crack_ca00558b}
```

picoCTF{atbash_crack_ca00558b}

Character Transformation Examples

k becomes p
r becomes i
x becomes c
l becomes o
X becomes C
G becomes T
U becomes F

FLAG

picoCTF{atbash_crack_ca00558b}

TOOLS USED

Steghide
Python 3

KEY TAKEAWAYS

- Challenge filenames often provide important hints about the intended solution.
- Empty passphrases are commonly used in beginner steganography challenges.
- The Atbash cipher is simple and symmetric, making it easy to implement.
- Combining steganography and cryptography is a common pattern in CTF challenges.

REFERENCES

Atbash Cipher – Wikipedia
Steghide Documentation
picoCTF Official Website

AUTHOR

Glenvio Regalito Rahardjo

Solved: December 2024

DISCLAIMER

This write-up is created for educational purposes only.
All techniques demonstrated follow CTF rules and ethical guidelines.