**BIOS**

**CTF-Challenges**

**Challenge 1**

**Multi Encoder**

**CHALLENGE DESCRIPTION**

**Its all about the base**

**Flag format - inctfj{flag\_here}**

**Approach**

I solved this challenge using the basic Linux commands. Initially I checked what is in the file given along with the task it is encoded text file using hex and binary also a python file explaining the encryption pattern. Then I checked the python file I found that it’s encrypted the file 5 times So then I took that encrypted file I understood that it is a hex encoded file then I used the following command to decode hex:  
cat ciper.txt| xxd -r -p | base64

In this the xxd -r -p it will convert them into binary afterwards it is encoded to base64 and stored to another txt file called ciper2.txt.

Then I decoded the base64 encoded file using the following command:

cat ciper2.txt | base64 --decode

it will decode the base64 encoded text.

I changed the content of txt file after each decryption.

I repeated this two commands several times to find the flag. Its not sure But I remember that if I am decoding hex once. I have to decode Base64 twice I repeated this several times and I found the flag.

**Challenge 2**

**base-base-base**

**CHALLENGE DESCRIPTION**

**Have you heard of bases? I heard combination of those can be really cool!**

**Approach**

In this task I solved with basic Linux commands. There is a text file attached to the file which contain the encoded text. When I opened the file I found the text is encoded with base 64 and I Used the following command to decode the base 64:

echo “encoded text” | base64 --decode

When it decoded it was found that the output text is encoded with base32 so I used following command to decode the base32 encoded text:

echo “encoded text” | base32 --decode

When it decoded it is found that the output text is encode with hex so afterwards I converted the hex to base64 for that I used the following command:

echo “hex encoded text” | xxd -r -p | base64

Then It will decode hex to base64 format Then I decoded the base64 encoded text using the following command:

echo “encoded text” | base64 --decode

After encoding the base64 I found he flag.

**Challenge 3**

**Single byte X0r**

**CHALLENGE DESCRIPTION**

**Given a hex encoded string: 1314190e1c1001024a0825194e145d0e251849251f4e091316032518084a11491407 The above string has been Xored against a single character ‘z’. Decode it to a meaningful sentence and get the flag.**

**Approach**

When I opened the task found a cipher text provided and found that it has been xored against a single character z. we just need to xor it again with the same key to get the flag.

**Challenge 4**

**Naïve Cipher**

**CHALLENGE DESCRIPTION**

**This Cipher is a very simple and common encryption method which forms part of the basis of cryptography. It simply shifts a string of letters a certain number of positions up or down the alphabet.**

**encoded string: dixoae{oczz\_ocz\_hvnozm\_ja\_xvznzm\_xdkczm}**

**Approach**

When I opened I found the encoded text and the text has been shifted with the key -5 by comparing the original flag to the text given .So we just decrypt it using this info.

**Challenge 5**

**Angry Steve**

**CHALLENGE DESCRIPTION**

**Jimmy found out about an embaraasing picture of his dad Steve. He believes that there is a text inside that file somwhere. But he is too scared to open and tamper with it as his father is a very short tempered man. Help Jimmy out!.**

**Approach**

I saw one image file attached to this task I downloaded it and found that some strings and unprintable character in it I used the following Linux command to find the flag.

Strings Angry.jpg | grep “inctf”

In this I used the string for extracting all the printable characters from the image file and used grep is used to search the keyword inctf from the specified file. Then I found the flag

**Challenge 6**

**Mischief Kid**

**CHALLENGE DESCRIPTION**

**Little Bart here is the biggest troublemaker in town. He is hiding the flag somewhere safe. Follow and bust out little Bart to get what you want!**

**Approach**

I found a zip file attached to the task. I downloaded the file and extract the file in it I found a image file in it some files are hided. So I tried to unzip that image file it extracted the files

In it I found one unsupported image file and a text file so I converted the text file into image file then I used a external tool called the ghex and I compared the hex of both image file then I found some are missing in one or them I added those and opened the image in it I found the flag.

**Challenge 7**

**Snow Man**

**CHALLENGE DESCRIPTION**

**Read the challenge name again! And yeah, everything you need is present in this file.**

**Approach**

I found a text file in attached to the task. I stored the content in a flag.txt. When I read the text I understood the password is thisiseasy I used an external tool called stegsnow the following command will help to find out the flag in base64 format:

Stegsnow -C -p ‘thisiseasy’ flag .txt

Then I found the base64 encoded text Then I decoded the base64 code using the following command to get the flag.

echo “encoded text” | base64 –decode