**Sartech Solutions Challenge Solution:  
Faith See (1002851)**

**Hint 0:** You don’t always apply what you learn in examinations, but today you do.

* Substitution + Transposition cipher

Second chance to actually get this right this time...

**Hint 2:** Take the start and the end of each sentence.

* Refer to step 1.

1. Taking the first letter and last letter of each line in the poem gives 2 keys, DICT and KILL.

*Death is not the end we seek*

*It is the beginning, for I*

*Cannot help but kneel*

*To the collosal hades’ will.*

1. The title of the poem is a ciphertext: PHIBFB8DKGEE

**Hint 3:** The beginning can be transposed but the end cannot be substituted.  
*(I could only understand this hint after solving the challenge by brute-forcing)*

* We can perform transposition at the start, but we cannot perform substitution at the end to obtain the deciphered text. (The deciphered text was obtained by performing substitution and transposition.)

1. Breaking the Vigenere cipher (substitution):
   1. Ciphertext: PHIBFB8DKGEE
   2. Key: KILL
   3. Plaintext: F97053X2A833

**Hint 1:** Key is written in hexadecimal and is 80 bits long. For example if PHIBFB8DKGEE is the ciphered key then 2 of the letters in the key is '0x'.

* Initial wrong interpretation: The key is 80 bits (10 characters) and so the key I am searching for would be ‘0x’ + 8 characters.
* Correct interpretation: The key is 80 bits (10 characters) and so the key I am searching for would be prefixed with ‘0x’, aka ‘0x’ + 10 characters.

1. Breaking the Transposition cipher:
   1. Ciphertext: F97053X2A833
   2. Key: DICT

|  |  |  |  |
| --- | --- | --- | --- |
| C | D | I | T |
| F97 | 053 | X2A | 833 |

|  |  |  |  |
| --- | --- | --- | --- |
| D | I | C | T |
| 0 | X | F | 8 |
| 5 | 2 | 9 | 3 |
| 3 | A | 7 | 3 |

Thus we can conclude that the hexadecimal key we are looking for is F852933A73.

Initially, I computed the decryption of almost all the Vigenere and Transposition Ciphers...manually. The last few I attempted I used an online decoder. Frankly felt quite silly after I realised that I was not being very smart about it, but I guess it was a good reminder to not get too caught up in the task at hand that I forget the bigger picture and final end goal.

|  |
| --- |
| python3 ecb.py -i random-1.png -o output-1.png -k F852933A73 -m d |

1. After running the code for all 7 given input files, we obtained [7 output files](https://drive.google.com/open?id=18mnRmePgQ097XfzRL45QC5eaHwMaaNAl) with MD5 hashes:

|  |
| --- |
| c337b66e9655a710d5a292bbe8365946 5578f65081824f19e2f253e59c91671f 2ed074586ca648092321eb284101350b ae7bc1462aec26b6eb44514af9041c76 9b76f2b398575b43d292063423931bb2 ee052027e2fded28c0e6acb12e2409cd 7100de6fd95ea13a2b32872371deeda7 |

1. Using an [MD5 hash cracker](https://hashcrack.com/), we obtained the following 7 hashes:

|  |
| --- |
| 2a535  15dic  49i37  524n9  o4258  ry560  363t9 |

Initially, I tried several other websites and the hashes could not be cracked. It was only until I tried the website I linked that the hashes could actually be cracked. Perseverance...is...key!!!

1. We combined the 7 hashes to obtain single string:

|  |
| --- |
| 2a53515dic49i37524n9o4258ry560363t9 |

**Hint 4:** The flag only contains alphabets.

1. Separating the numbers from the alphabets, we obtained 2 strings:

|  |
| --- |
| adicinoryt 2535154937524942585603639 |

1. *Bonus step because Faith is...not so bright. I thought that I needed to combine the 7 hashes to form 1 hash, but I realised that there were too many characters. After stripping the alphabets from the hash, I noticed there were 25-characters. Considering that I needed a 32-character long MD5 hash to crack, I thought that I needed to combine the 25-characters with the order in which the 7 hashes were used in forming the word dictionary (I didn’t think of other possible permutations).  
     
   Here are the hashes that I tried, and obviously failed to crack.  
     
   This is the order in which the hashes appeared to form the word ‘dictionary’:*

|  |
| --- |
| 15dic363t949i37o4258524n92a535ry560 |

*After splitting the alphabets from the numbers:*

|  |
| --- |
| dictionary  1536394937425852492535560 |

*The order is 2-7-3-5-4-1-6:*

|  |
| --- |
| order = 2735416  numbers\_dictionary = 1536394937425852492535560  numbers\_original = 2535154937524942585603639  27354161536394937425852492535560 (order + numbers\_dictionary) 15363949374258524925355602735416 (numbers\_dictionary + order) 15236397493734258552494253515606 (numbers\_dictionary with the order appended to the end   of each individual hash) 15236379493375425852449215356560 (numbers\_dictionary with the order appended in each   individual hash at the point where the alphabets were) 25351549375249425856036392735416 (numbers\_original + order) 27354162535154937524942585603639 (order + numbers\_original) |

This really taught me...to try my best not to overthink and over complicate situations. The answer was right there, but I was so convinced it was “harder”.

1. Running the alphabets through an anagram solver, there are 2 possible flags:

|  |
| --- |
| dictionary indicatory |

**Final flag:** CTF{indicatory}

Initially, I assumed that the flag had to be CTF{dictionary}. This meant that I kept rechecking my spelling over and over without considering that it could be something else. Once again, a reminder to not forget that there are other potential solutions...