

Digital defenders challenge writeup

- Bikramaditya Munshi

Username :- bikramadityamunshi

Email: - bikrammunshi@hotmail.com

1. Cryptography challenge : x0rbash

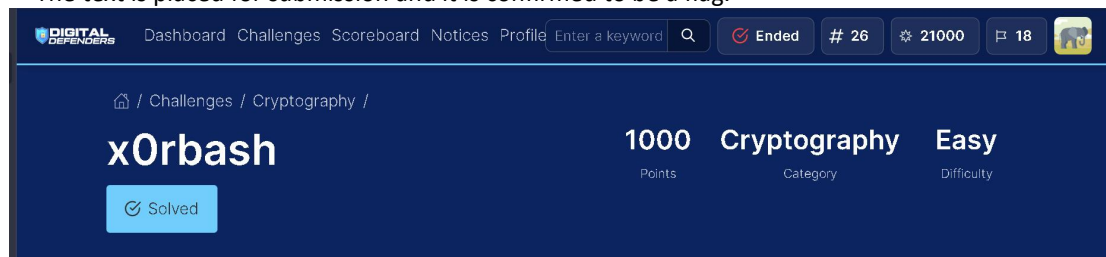
One output.txt file and another xorbash.py file is provided.

On analyzing the python code it is noticed that the plain text is encoded in an affine cipher specified in the program. The key and the affine cipher is converted to bytes using utf-8 encoding. Then each byte of the cipher is raised to the power of the value provided by the nth byte of the key. Where n is the position of the byte of the cipher % the length of the key. The new cipher text is then decoded from utf-8 encryption and encoded using base64 encryption to give the cipher text.

The output.txt file contains this encrypted cipher text.

The cipher text is then decoded first from base64 encryption and encoded using utf-8 encryption. The key is also encrypted using utf-8 encryption. Then each byte of the cipher is raised to the power of the value provided by the nth byte of the key. Where n is the position of the byte of the cipher % the length of the key. Then the new cipher is decoded using utf-8 decryption and the affine cipher method is applied. The result gives a text which is in the flag format.

The text is placed for submission and it is confirmed to be a flag.



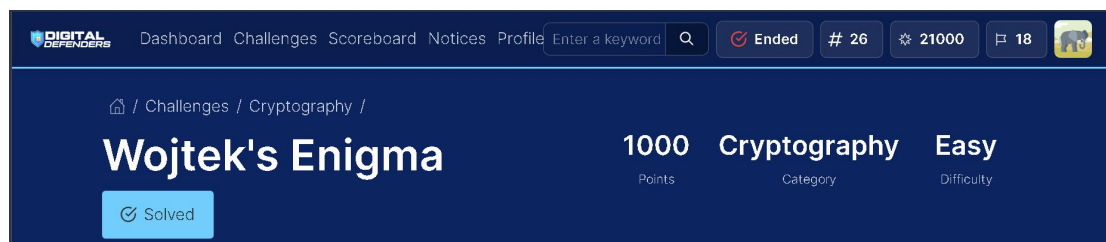
2. Cryptography challenge : Wojtek's enigma

One file chall.txt is provided.

The file contains a string which looks like a flag but encoded.

Some model, reflector, rotor1, position, ring, rotor2, position, ring, rotor3, position, ring and plugboard information is provided.

From the challenge name it is understood enigma machine is used for encryption. So we need an enigma decryption machine. These are available online the information provided has to be entered in the appropriate places and the string is decrypted to give us the flag.

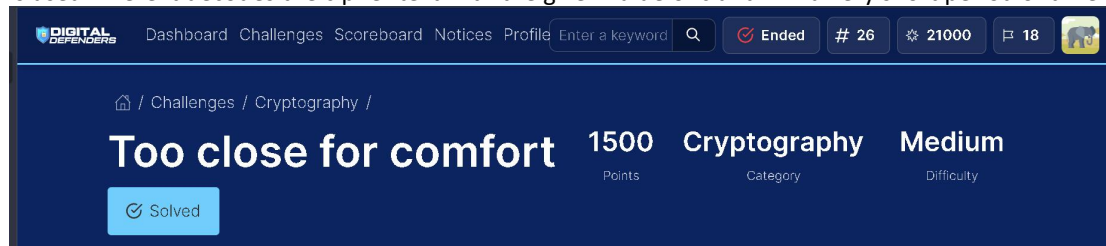


3. Cryptography challenge : Too close for comfort

Two files chall.py and output.txt are provided.

The text file contains a variable n and a variable c with their values.

Analyzing the python script, it is understood a rsa encryption is used where n is the modulus N and c is the cipher text. To decode we need the value of d for which we need to find the prime factors of n . N being a very large number instead of going the old way, a tool available in github called RsaCtfTool is used where it decodes the cipher text with the given value of c and n in a very short period of time.



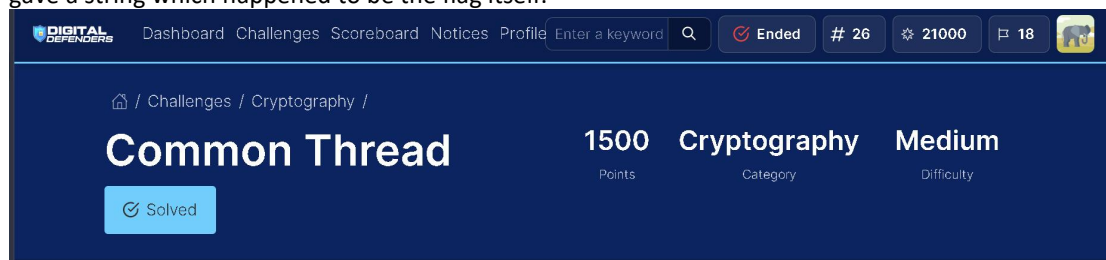
4. Cryptography challenge : Common Thread

Two files, chall.py and output.txt are provided.

The text file provides us with the values of n and $ct1$ and $ct2$

The python file provides us with a rsa encryption code with a catch. The catch being there are 2 values of e which encodes the string into two cipher texts $ct1$ and $ct2$.

A thought arised and the value of $ct1$, $e1$ and n was provided to the RsaCtfTool and it decoded and gave a string which happened to be the flag itself.

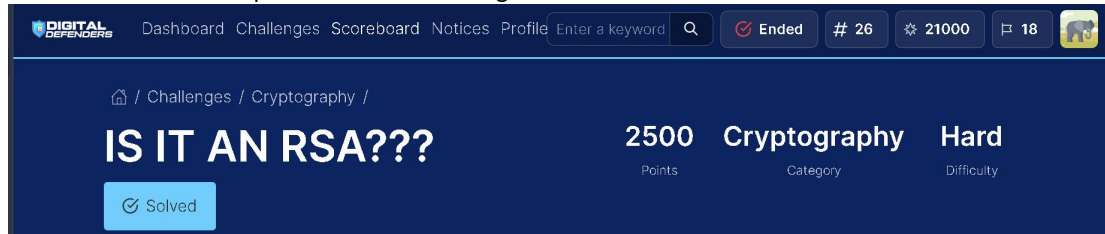


5. Cryptography challenge : Is It An Rsa???

Two files are provided : chall.py and cipher.txt

The text provides us with three values of n (n1,n2,n3) and a cipher text.

The python file provides us with the encryption code of rsa cipher. The cipher text seems to be encrypted with n3 as modulus. Hence the value of n3 is provided in the RsaCtfTool with the cipher text and it decoded to provide us with the flag.



6. Network Security challenge : One_by_one

A pcapng file is provided with the name One_by_one.

In the challenge description it is mentioned every packet has a character in its payload.

The pcapng file is loaded into wireshark and payload of each packets were checked. The characters found were noted down in one place and at the end the string gave rise to something resembling a flag, its tried out and it is the flag indeed.



7. Network Security challenge : Packet_Sniffing

A pcapng file with the name Packet_Sniffing is provided.

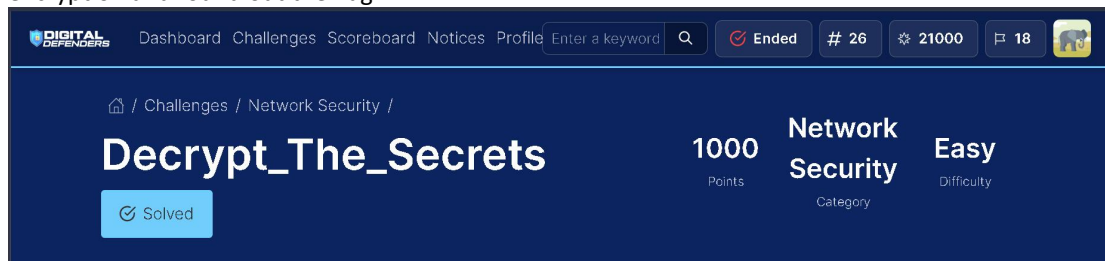
There are many protocols in the file. There is a http protocol where the payload resembles a jpg image file. Copied the raw data and using the bless hexeditor a jpg is created from the raw data. The flag is given in the picture.



8. Network Security challenge : Decrypt_The_Secrets

A pcapng file named Decrypt_The_Secrets is provided.

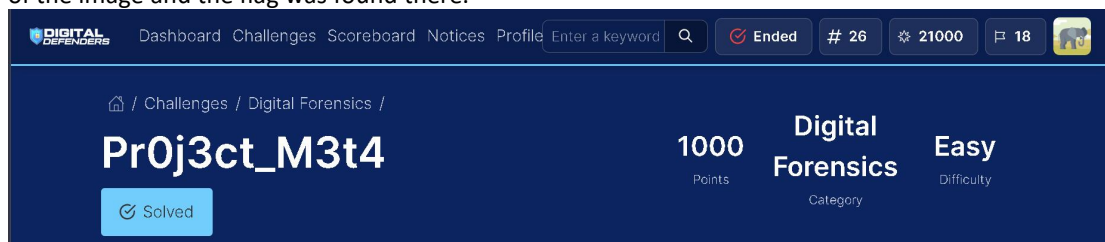
The flag is provided in an encoded form within the payloads of the packets. It is not a base64 encoding because base64 encoding has = at the end. So tried the next famous encryption the rot13 encryption and found out the flag.



9. Digital Forensics : Pr0j3ct_M3t4

A chall.jpg image file is provided.

The challenge name suggests something about meta, so exiftool is used to get the meta information of the image and the flag was found there.

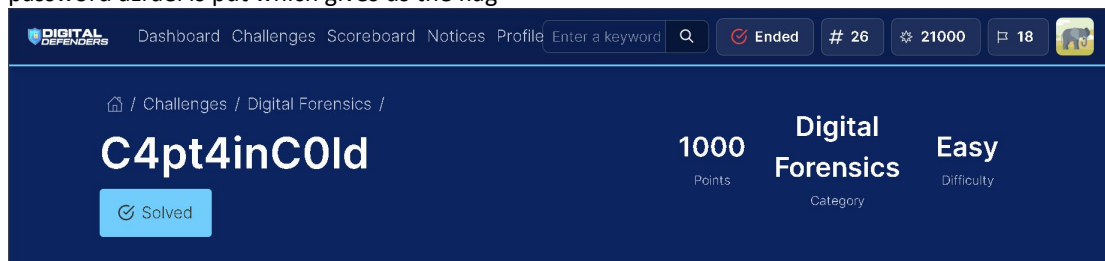


10. Digital Forensics : C4pt4inC0ld

A secret.txt is provided.

The text file contains some text where one interesting thing is 'azrael' which is said to be a password of some sort.

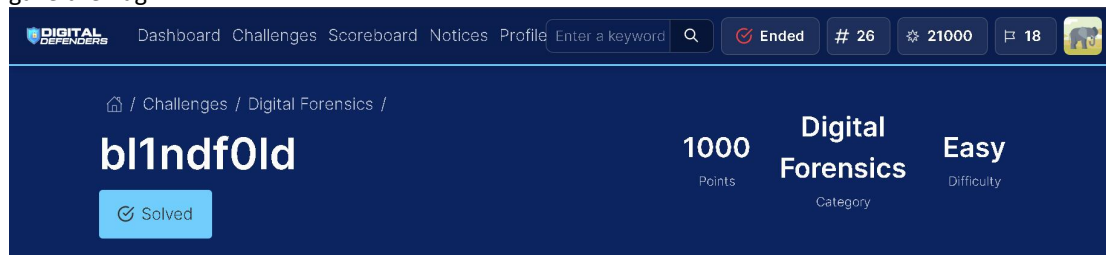
A steganography tool is used to extract hidden data from the txt file. A popup appears asking for password azrael is put which gives us the flag



11. Digital Forensics : bl1ndf0ld

A png file of the name bl1ndf0ld is provided in the challenge.

Image appears to be blank in plain sight. So stegsolve is applied where there is an option for steganography solver which gives us certain channels to move through. Moving through all channels gave the flag.



12. Web challenge : Partly_stored_answers

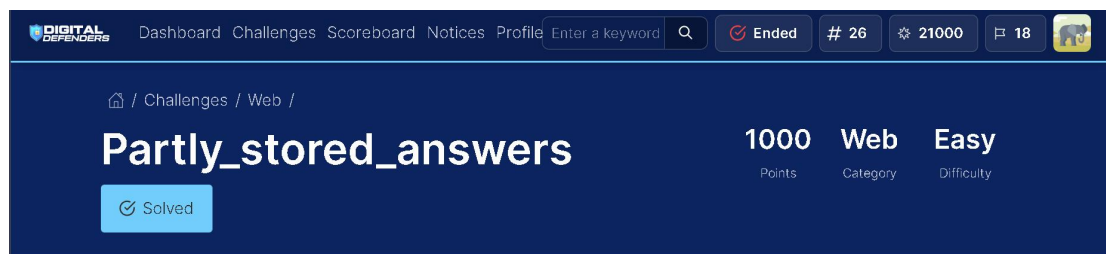
First the source code is checked. The first part of the flag is given there.

The challenge descriptions mentions something about robots. We know there is a robots.txt file for web pages that tells the search engine crawlers which URL the crawlers can access. So /robots.txt is applied in the url which on showed a different page. The page displays '/secrets/k3y' and on inspection gives the second part of the key.

The '/secrets/k3y' domain is visited which from naked eye shows nothing valuable but on inspection gives something called 'secret' with a value on it.

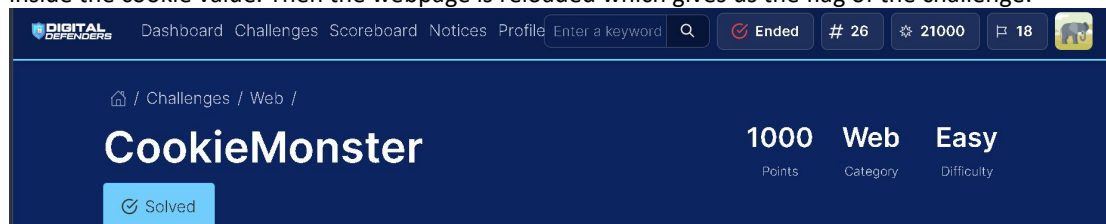
The original webpage has a text field wanting for secret, the value which secret from 'secrets/k3y' provided is entered which gives us the final part of the flag.

All parts are combined to get the flag for the challenge.



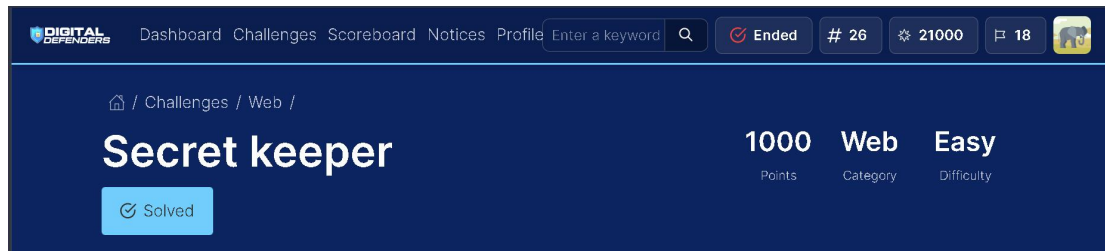
13. Web challenge : CookieMonster

The webpage shows about among us. On inspection the cookie value shows something about a url encoding, the url encoded is decoded which give a base64 encryption which on decoding shows admin value 0. The value of admin is changed to 1 , base64 encrypted then url encoded and is put inside the cookie value. Then the webpage is reloaded which gives us the flag of the challenge.



14. Web challenge : Secret Keeper

Tried for SQL injection with parameter ' or 1=1-- ' in username and password. It gave the flag.



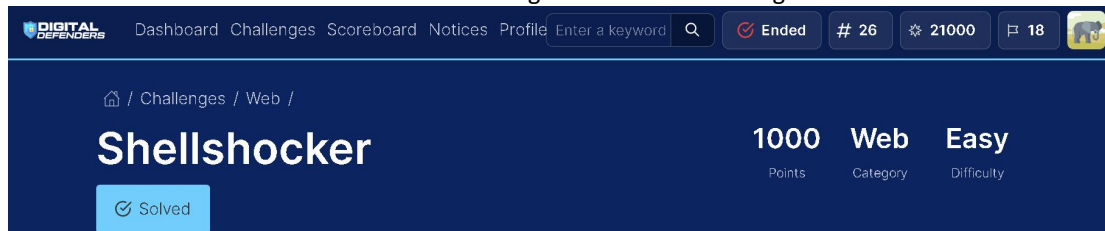
15. Web challenge : Phone Book

The web page has a click here button. When the button is clicked it takes to a new web page where user information is provided. The account id in the url shows 1. This value is changed to 0 which gives us the flag.



16. Web challenge : Shellshocker

Using postman to search for directories. Searched through every directory and found '../..//flag.txt'. Used the command cat command for the file to get the value of the flag.




17. Web challenge : Xml Parser

Found XXE vulnerability. Tried XXE injection code.

```
<!--?xml version="1.0" ?-->
<!DOCTYPE replace [<!ENTITY ent SYSTEM "file:///etc/shadow"> ]>
<userInfo>
  <firstName>John</firstName>
  <lastName>&ent;</lastName>
</userInfo>
```

Pasted the code and submitted which gave the flag.

DashboardChallengesScoreboardNoticesProfileEnter a keyword🔍

Ended

26

21000

18



[/ Challenges / Web /](#)

Xml Parser

Solved

15001500Points

WebWebCategory

MediumMediumDifficulty

18. Web challenge : Laughable File Infiltration

The website itself provides no valuable information.

The website requests are intercepted using Burpsite and the proxy signal is repeated several times to send signals to the server as the website. This gives a possible location which end as /flag.txt . The given domain is visited which gives the flag for the challenge.


DashboardChallengesScoreboardNoticesProfileEnter a keyword🔍

Ended

26

21000

18



[/ Challenges / Web /](#)

Laughable File Infiltration

Solved

10001000Points

WebWebCategory

EasyEasyDifficulty