vishwaCTF

CHALLENGE NAME : THE NAUGHTY FRIEND

DEV : KANISHK KUMAR

CATEGORY : CRYPTOGRAPHY

LEVEL : HARD

2024

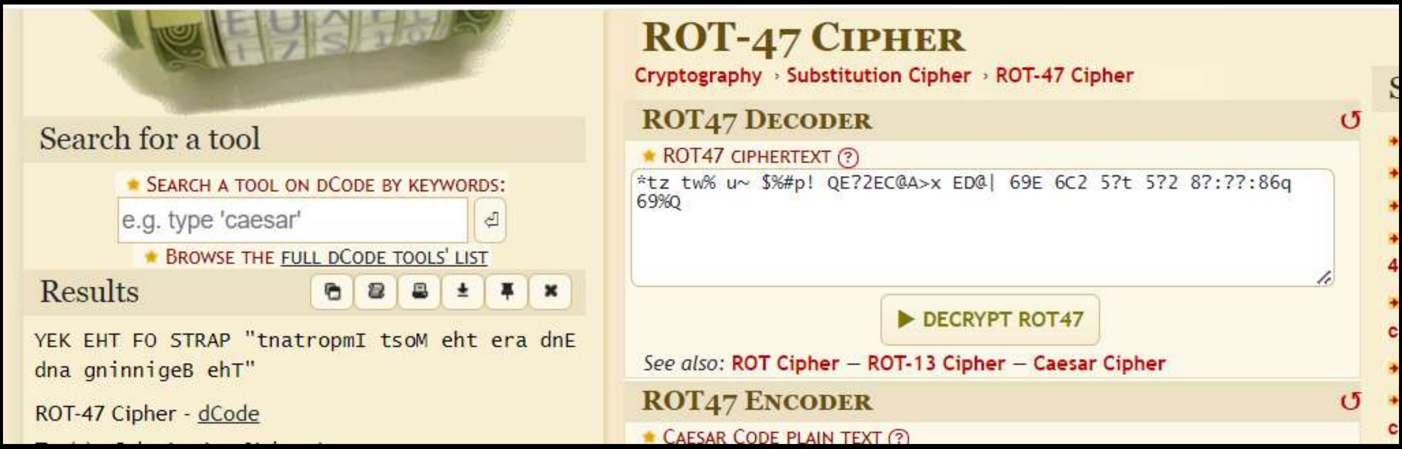
Question Description:

One of my friends Dhruv is a cryptography genius, but he likes to annoy me by playing pranks with my passwords. He recently changed my accounts password and has given the following files as hints, he also gave this buggy code which had some import statements removed, help me retrieve my lost password!!!

Solution:

Step 1:

Firstly, Decrypt the key using any ROT-47 decoder:


Search for a tool
SEARCH A TOOL ON DCODE BY KEYWORDS:
e.g. type 'caesar'
BROWSE THE FULL DCODE TOOLS' LIST
Results
ROT-47 CIPHER
Cryptography › Substitution Cipher › ROT-47 Cipher
ROT47 DECODER
ROT47 CIPHERTEXT
DECRYPT ROT47
See also: ROT Cipher – ROT-13 Cipher – Caesar Cipher
ROT47 ENCODER
CAESAR CODE PLAIN TEXT

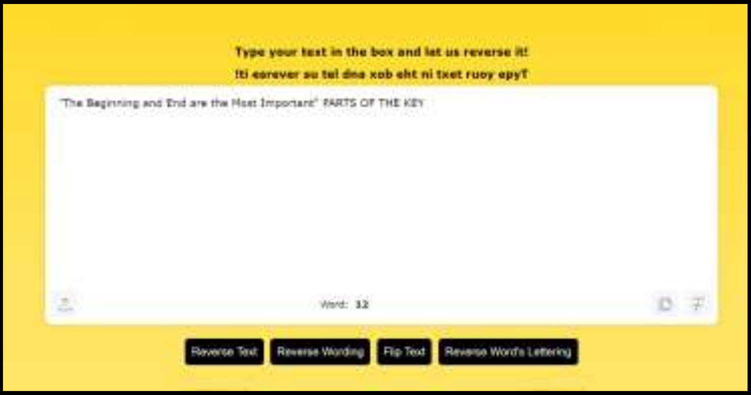
```
*tz tw% u~ $%#p! QE?2EC@A>x ED@| 69E 6C2 5?t 5?2 8?:??:86q
69%Q
```

```
YEK EHT FO STRAP "tnatropmI tsoM eht era dnE
dna gninnigeB ehT"
```

ROT-47 Cipher - dCode

You will get the text: YEK EHT FO STRAP "tnatropmI tsoM eht era dnE dna gninnigeB ehT"

You can notice that the text has been reversed, so you can use any text reverser online to get the original text which is : "The Beginning and End are the Most Important" PARTS OF THE KEY


Type your text in the box and let us reverse it!
!ti esrever su tel dna xob eht ni txet ruoy epyT
Word: 12
Reverse Text
Reverse Wording
Flip Text
Reverse Word's Lettering

"The Beginning and End are the Most Important" PARTS OF THE KEY

From this you can interpret that we have to take the beginning and the end letters of "The Beginning and End are the Most Important" .

So the key would be:

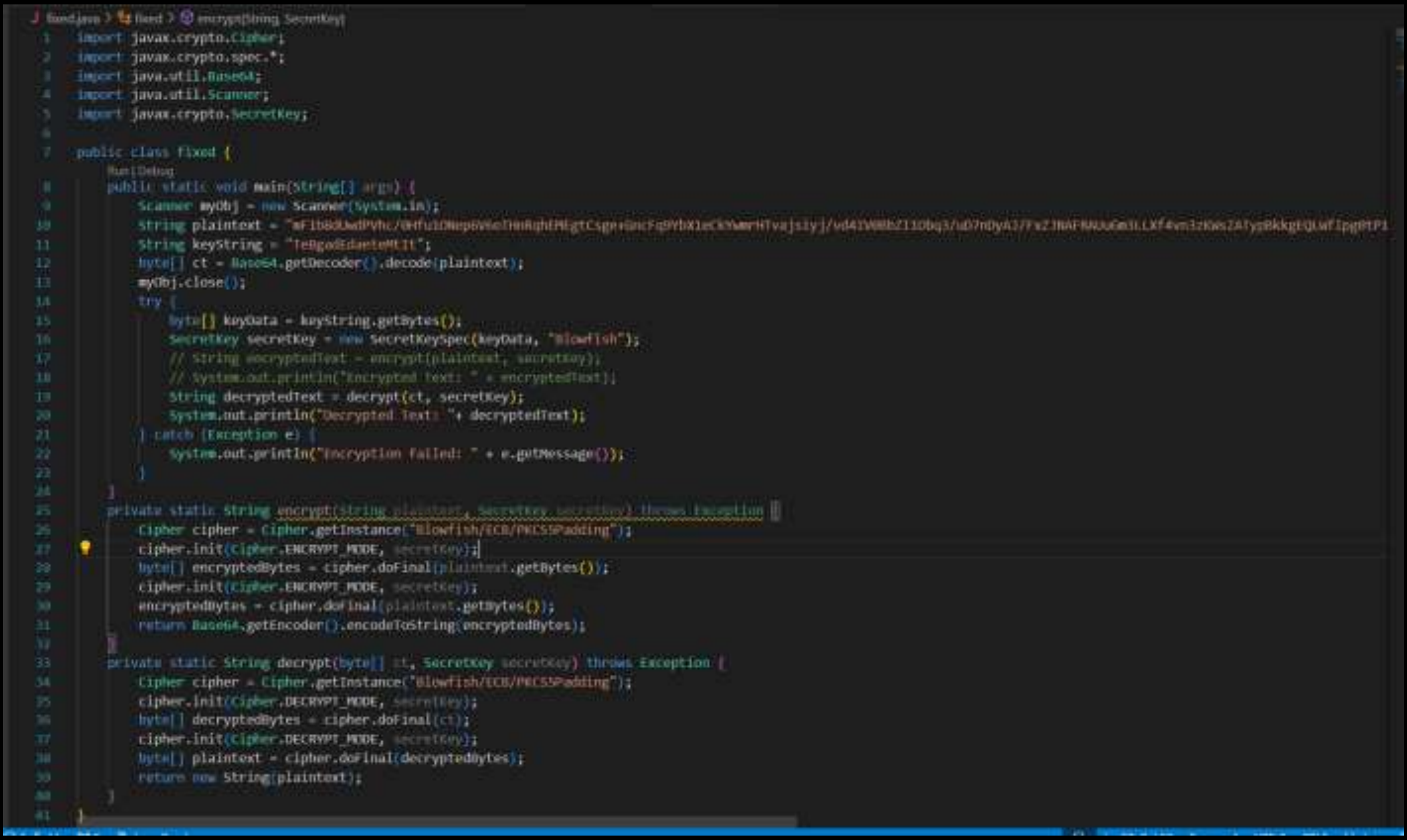
TeBgadEdaeteMtIt

Step 2:

Firstly we have to fix the code by importing java.security , java.crypto.Cipher and catch exception.

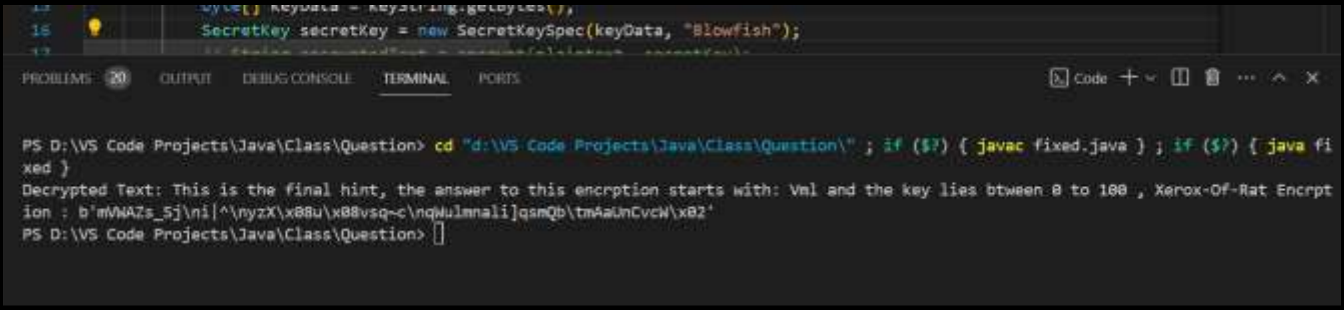
The Fixed code will be like:

```java
import javax.crypto.Cipher;
import javax.crypto.spec.*;
import java.util.Base64;
import java.util.Scanner;
import javax.crypto.SecretKey;

public class fixed {
    public static void main(String[] args) {
        Scanner myObj = new Scanner(System.in);
        String plaintext = "mF1b8dUwdPVhc/0Hfu1ONep6V6oTH-
nRqhEMEgtCsge+GncFq9YbX1eCkYwmrHTvajsiyj/vd4IV0BbZI1Obq3/uD7nDyAJ/FxZJNAFRAU-
uGm3LLXf4vn3zKWsZATypBkkgEQLWfIpg0tP13wJRhk6JUVPi17AaKHrodTt-
WOq54FqKIaT1DoifMjtJ4TCG3IXmjEo+6ZsBokIjxeCjamGBwNAqFaqIik-
kHJo7L1PiCFds/lAaB38KqHGL/E2pfw0CK3XYzKV8gBdwhnrUq1UN1Q";
        String keyString = "TeBgadEdaeteMtIt";
        byte[] ct = Base64.getDecoder().decode(plaintext);
        myObj.close();
        try {
            byte[] keyData = keyString.getBytes();
            SecretKey secretKey = new SecretKeySpec(keyData, "Blowfish");
            // String encryptedText = encrypt(plaintext, secretKey);
            // System.out.println("Encrypted Text: " + encryptedText);
            String decryptedText = decrypt(ct, secretKey);
            System.out.println("Decrypted Text: "+ decryptedText);
        } catch (Exception e) {
            System.out.println("Encryption failed: " + e.getMessage());
        }
    }
    private static String encrypt(String plaintext, SecretKey secretKey) throws Exception
{
        Cipher cipher = Cipher.getInstance("Blowfish/ECB/PKCS5Padding");
        cipher.init(Cipher.ENCRYPT_MODE, secretKey);
        byte[] encryptedBytes = cipher.doFinal(plaintext.getBytes());
        cipher.init(Cipher.ENCRYPT_MODE, secretKey);
        encryptedBytes = cipher.doFinal(plaintext.getBytes());
        return Base64.getEncoder().encodeToString(encryptedBytes);
    }
    private static String decrypt(byte[] ct, SecretKey secretKey) throws Exception {
        Cipher cipher = Cipher.getInstance("Blowfish/ECB/PKCS5Padding");
        cipher.init(Cipher.DECRYPT_MODE, secretKey);
        byte[] decryptedBytes = cipher.doFinal(ct);
        cipher.init(Cipher.DECRYPT_MODE, secretKey);
        byte[] plaintext = cipher.doFinal(decryptedBytes);
        return new String(plaintext);
    }
}
```



```
fixed.java > fixed > encrypt(String, SecretKey)
import javax.crypto.Cipher;
import javax.crypto.spec.*;
import java.util.Base64;
import java.util.Scanner;
import javax.crypto.SecretKey;

public class fixed {
    Run|Debug
    public static void main(String[] args) {
        Scanner myObj = new Scanner(System.in);
        String plaintext = "mF1b8dUwdPVhc/0Hfu1DNep6VkoTHnRqhEMEgtCsge+GncFq9YbX1eCkYwmrHTvajsiyj/vd41W8bZ11Dbq3/uD7nDyA3/FxZJNAFNAUuGm3LLXf4vn3zKWs2ATypBkkgEQLwfIpg8tP1
        String keyString = "TeBgodEdaeteMtIt";
        byte[] ct = Base64.getDecoder().decode(plaintext);
        myObj.close();
        try {
            byte[] keyData = keyString.getBytes();
            SecretKey secretKey = new SecretKeySpec(keyData, "Blowfish");
            // String encryptedText = encrypt(plaintext, secretKey);
            // System.out.println("Encrypted text: " + encryptedText);
            String decryptedText = decrypt(ct, secretKey);
            System.out.println("Decrypted Text: "+ decryptedText);
        } catch (Exception e) {
            System.out.println("Encryption Failed: " + e.getMessage());
        }
    }
    private static String encrypt(String plaintext, SecretKey secretKey) throws Exception {
        Cipher cipher = Cipher.getInstance("Blowfish/ECB/PKCS5Padding");
        cipher.init(Cipher.ENCRYPT_MODE, secretKey);
        byte[] encryptedBytes = cipher.doFinal(plaintext.getBytes());
        cipher.init(Cipher.ENCRYPT_MODE, secretKey);
        encryptedBytes = cipher.doFinal(plaintext.getBytes());
        return Base64.getEncoder().encodeToString(encryptedBytes);
    }
    private static String decrypt(byte[] ct, SecretKey secretKey) throws Exception {
        Cipher cipher = Cipher.getInstance("Blowfish/ECB/PKCS5Padding");
        cipher.init(Cipher.DECRYPT_MODE, secretKey);
        byte[] decryptedBytes = cipher.doFinal(ct);
        cipher.init(Cipher.DECRYPT_MODE, secretKey);
        byte[] plaintext = cipher.doFinal(decryptedBytes);
        return new String(plaintext);
    }
}
```

Giving This as Output:



```
            SecretKey secretKey = new SecretKeySpec(keyData, "Blowfish");
```

```
PS D:\VS Code Projects\Java\Class\Question> cd "d:\VS Code Projects\Java\Class\Question\" ; if ($?) { javac fixed.java } ; if ($?) { java fi
xed }
Decrypted Text: This is the final hint, the answer to this encrption starts with: Vml and the key lies btween 0 to 100 , Xerox-Of-Rat Encrpt
ion : b'mVWAZs_Sj\ni|^\nyzX\x08u\x08vsq~c\nqWulmnali]qsmQb\tmAaUnCvcW\x02'
PS D:\VS Code Projects\Java\Class\Question> []
```

Decrypted Text: This is the final hint, the answer to this encrption starts with: Vml and the key lies btween 0 to 100 , Xerox-Of-Rat Encrption : b'mVWAZs_Sj\ni|^\nyzX\x08u\x08vsq~c\nqWulmnali]qsmQb\tmAaUnCvcW\x02'

Step 3:

From the given Output we get the hints that encryption used is XOR encryption and the key lies between 0 to 100. Also the text will start with "Vml".

Encrypted Bytes Would be:

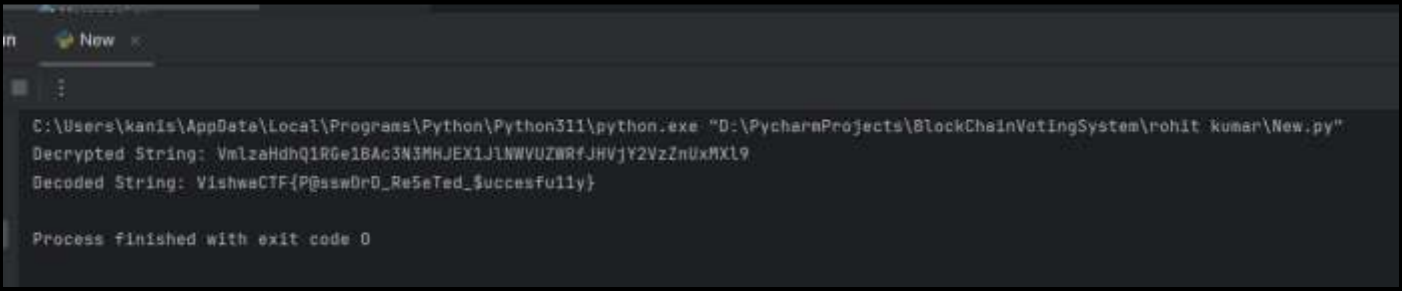
```
b'mVWAZs_Sj\ni|^\nyzX\x08u\x08vsq~c\nqWulmnali]qsmQb\tmAaUnCvcW\x02'
```

The encrypted bytes can be decrypted using a simple python code for brute forcing XOR encryption.

Code:

```python
from pwn import xor
import base64

# DECRYPTION

# Given encrypted byte string
encrypted_bytes = b'mVWAZs_Sj\ni|^\nyzX\x08u\x08vsq~c\nqWulmnali]qsmQb\tmAaUnCvcW\x02'

# Brute-force decryption using XOR cipher for keys 1 to 100
for key in range(1, 101):
    # Perform decryption using the XOR cipher
    decrypted_bytes = bytearray()
    for byte in encrypted_bytes:
        decrypted_byte = byte ^ key
        decrypted_bytes.append(decrypted_byte)

    # Convert decrypted bytes to string
    decrypted_string = decrypted_bytes.decode()
    length = len(decrypted_string)

    if 'Vml' in decrypted_string:
        print("Decrypted String: "+decrypted_string)
        decoded_string = base64.b64decode(decrypted_string).decode()
        print("Decoded String: " + decoded_string)
        break
```

Output:



```
C:\Users\kanis\AppData\Local\Programs\Python\Python311\python.exe "D:\PycharmProjects\BlockChainVotingSystem\rohit kumar\New.py"
Decrypted String: VmlzaHdhQ1RGe1BAc3N3MHJEX1JlNWVUZWRfJHVjY2VzZnUxMXl9
Decoded String: VishwaCTF{P@ssw0rD_Re5eTed_$uccesfu11y}

Process finished with exit code 0
```

Flag:

VishwaCTF{P@ssw0rD_Re5eTed_$uccesfu11y}