Hgame week3 WP

RE

Arithmetic

```python
def max_path_sum_with_path(pyramid):
    n = len(pyramid)

    # 创建 dp 数组并初始化最后一行为数字金字塔最后一行
    dp = [[0] * (i + 1) for i in range(n)]
    dp[n - 1] = pyramid[n - 1]

    # 创建路径记录数组
    paths = [[-1] * (i + 1) for i in range(n)]
    paths[n - 1] = [j for j in range(n)]

    # 逐行向上计算 dp 数组和记录路径
    for i in range(n - 2, -1, -1):
        for j in range(i + 1):
            if dp[i+1][j] > dp[i+1][j+1]:
                dp[i][j] = pyramid[i][j] + dp[i+1][j]
                paths[i][j] = j
            else:
                dp[i][j] = pyramid[i][j] + dp[i+1][j+1]
                paths[i][j] = j + 1

    # 回溯得到路径
    num=[]
    path = []
    pos = 0
    for i in range(n - 1):
        path.append(pyramid[i][pos])
        if pos == paths[i][pos]:
           num.append(1)
        if pos!=paths[i][pos]:
            num.append(2)
        pos = paths[i][pos]
    path.append(pyramid[n - 1][pos])

    # 返回最大路径和和路径
    return dp[0][0], path,num
```

```
    pyramid =[txt中的数组]
    result_sum, result_path,num = max_path_sum_with_path(pyramid)
print("最大路径和:", result_sum)
print("路径:", result_path)
print("num:",num)
```

跑出即可获得1和2组成的数据md5后即可获得flag

babyre

开启了很多个线程，动态调试可以知道4次为一轮进行循环加密

```
sem_init(&stru_55EF729052A0, 0, 0);
sem_init(&stru_55EF729052C0, 0, 0);
pthread_create(&newthread, 0LL, first, 0LL);
pthread_create(&v7, 0LL, second, 0LL);
pthread_create(&v8, 0LL, five, 0LL);
pthread_create(v9, 0LL, third, 0LL);
for ( j = 0; j <= 3; ++j )
  pthread_join(*(&newthread + j), 0LL);
```

顺序依次往下，已知最后一位是}可以写脚本进行逆推

```
include<stdio.h>
int main()
{
        unsigned int input1[] = { 0x00002F14, 0x0000004E, 0x00004FF3, 0x0000006D, 0x000032D8, 0x0000006D, 0x00006B4B, 0xFFFFFF92, 0x0000264F, 0x0000005B, 0x000052FB, 0xFFFFFF9C, 0x00002B71, 0x00000014, 0x00002A6F, 0xFFFFFF95, 0x000028FA, 0x0000001D, 0x00002989, 0xFFFFFF9B, 0x000028B4, 0x0000004E, 0x00004506, 0xFFFFFFDA, 0x0000177B, 0xFFFFFFFC, 0x000040CE, 0x0000007D, 0x000029E3, 0x0000000F, 0x00001F11, 0x000000FF,0xF9};
        unsigned char key[] = {0x77, 0x74, 0x78, 0x66, 0x65, 0x69 };
        int input11 = 0x68;
        int input22 = 0x67;
        for (int routn = 31; routn >=0; routn--)
        {
                input1[routn] ^= (input1[routn + 1] - key[(routn+1 ) % 6]);
        routn--;
                input1[routn] /= (input1[routn+1] + key[(routn+1)%6]);
                routn--;
                input1[routn] += key[(routn+1)%6] ^ input1[routn+1];
                routn--;
                input1[routn] -= key[(routn+1)%6] * input1[routn+1];
        }

        for (int i = 0; i < 32; i++)
        {
                printf("%c", input1[i]);
```

```
        }
        return 0;
}
```

hgame{you_are_3o_c1ever2_3Olve!}

babyAndroid

用户名使用RC4加密，并且用户名传入本地调用进行AES加密

```
public static int key = 2131689520;
/* JADX INFO: Added by JADX */
public static final int abc_action_bar_home
```

ID指向2131689520

```
<string name="key">3e1fel</string>
```

找到真key进行RC4解密得到用户名然后再解开AES即可。

ezcpp

使用了tea算法并且每次移一字节有循环自加密逆推即可

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
void decypt(unsigned int* enc)
{
        unsigned int enc1 = enc[0], enc2 = enc[1];
        unsigned int detle = 0xDEADBEEF,sum = 32 * detle ;
        for (int i = 0; i < 32; i++)
        {
                enc2 -= (sum + enc1) ^ (16 * enc1 + 3412) ^ (32 * enc1 + 4123);
                enc1 -= (sum + enc2) ^ (16 * enc2 + 1234) ^ (32 * enc2 + 2341);
                sum -= detle;
        }
        enc[0] = enc1;
        enc[1] = enc2;
}
int main()
{
   char enc[]= {
0x88, 0x6A, 0xB0, 0xC9, 0xAD, 0xF1, 0x33, 0x33, 0x94, 0x74, 0xB5, 0x69, 0x73, 0x5F, 0x30, 0x62,
 0x4A, 0x33, 0x63, 0x54, 0x5F, 0x30, 0x72, 0x31, 0x65, 0x6E, 0x54, 0x65, 0x44, 0x3F, 0x21, 0x7D
  };
```

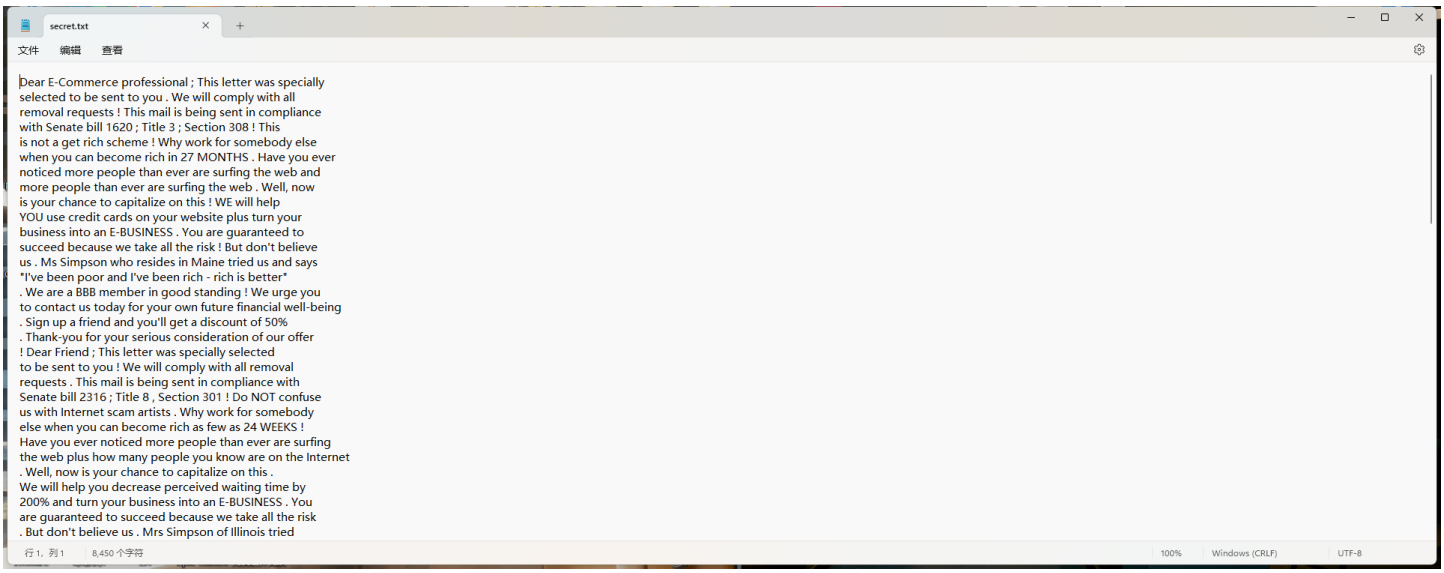
```
 char* p = enc+4;
  for (int n = 0; n <4; n++)
  {
          p--;
          decypt((unsigned int*)p);
  }
  for (int i = 0; i < 32; i++)
  {
          printf("%c", enc[i]);
  }
        return 0;
}
```

hgame{#Cpp_is_0bJ3cT_0r1enTeD?!}

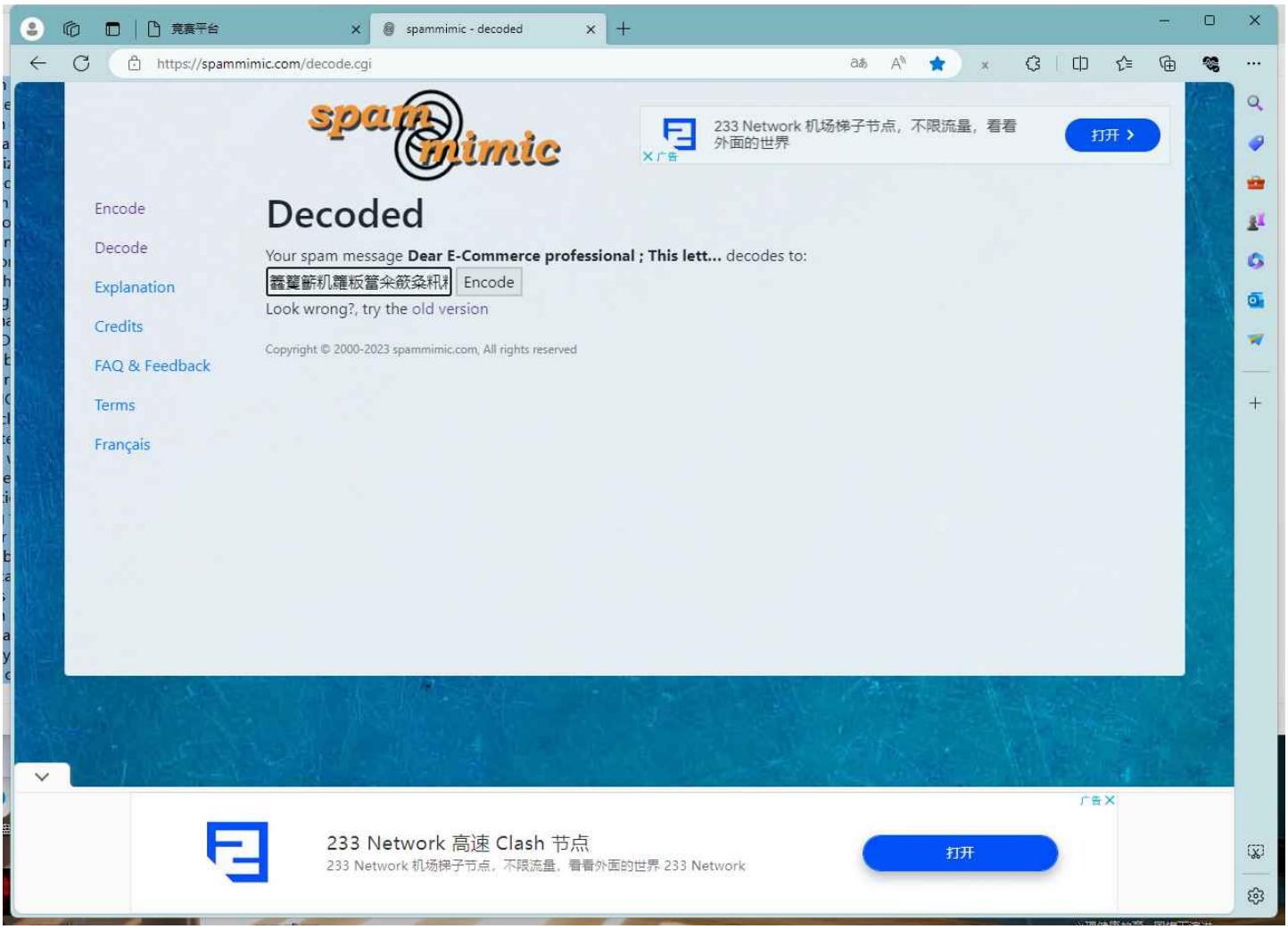
MISC

ezWord

解压word文档，拿到两张图片，用bwm.py解开水印得到压缩包密码



拿到英文文档，垃圾邮箱在线解密



通过猜想查找每个字符的unicode编码，发现unicode-9即为flag