

HGAME 2022 Week1 writeup by sasasas

HGAME 2022 Week1 writeup by sasasas

CRYPTO

Easy RSA

English Novel

IoT

饭卡的uno

MISC

欢迎欢迎

这个压缩包有点麻烦

好康的流量

群青(其实是幽灵东京)

Web

easy_auth

Tetris plus

蛛蛛

REVERSE

easyasm

Flag Checker

CRYPTO

Easy RSA

1. 百度"RSA"
2. 记事本打开文件, 发现38组数字
3. b站找求RSA工具
4. 将38组数字解出来, 转成字符, 得flag

English Novel

1. 下载得压缩包并解压
2. 记事本打开flag.enc, 发现一串密文
3. 记事本打开encrypt.py, 发现加密规则, 但缺少密钥

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
def encrypt(data, key):
    assert len(data) <= len(key)
    result = ""
    for i in range(len(data)):
        if data[i].isupper():
            result += chr((ord(data[i]) - ord('A') + key[i]) % 26 + ord('A'))
        elif data[i].islower():
            result += chr((ord(data[i]) - ord('a') + key[i]) % 26 + ord('a'))
        else:
            result += data[i]
    return result
```

- 4.同时打开文件夹"original"和"encrypt",并将各自的.txt文件按"字节大小"排序
- 5.筛选一：寻找两个文件夹中字节大小相同的文件(最好该字节大小一个文件夹里唯一)
- 6.筛选二：打开文件看看空格和字符是否对的上
- 7.根据加密前后的文件，写程序暴力枚举每一位密钥
- 8.找到四五组文件后，凑齐能够解密flag的key[],并解密得flag

IoT

饭卡的uno

- 1.ida 以“Intel Hex”打开文件
- 2.在hex窗口往下翻，发现flag

05 90 F4 91 E0 2D 09 94 F8 94 FF CF 00 00 00 00
FF 00 5F 00 8C 00 4C 01 BD 00 9B 00 AF 00 68 67	.._...L.....hg
61 6D 65 7B 46 31 72 73 74 5F 35 74 65 70 5F 30	ame{F1rst_5tep_0
46 5F 49 4F 54 7D 00 0D 0A 00 ?? ?? ?? ?? ?? ??	F_IOT}....??????
?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??	????????????????

MISC

欢迎欢迎

- 1.关注公众号

这个压缩包有点麻烦

- 1.winhex打开文件，末尾有提示"6位及以下纯数字"，用ARCHPR暴力破解第一个压缩包
- 2.用第一个压缩包里的txt作字典，ARCHPR破解第二个压缩包密码

- 3.发现满足明文破解条件，所以ARCHPR明文攻击第三个压缩包
- 4.将.jpg文件用winhex打开，搜索FFD9(jpg文件结尾)，发现后面还有504B(zip开头)。把压缩包从图片里拿出来
- 5.winhex打开压缩包，尝试伪密码，成功，得到flag

好康的流量

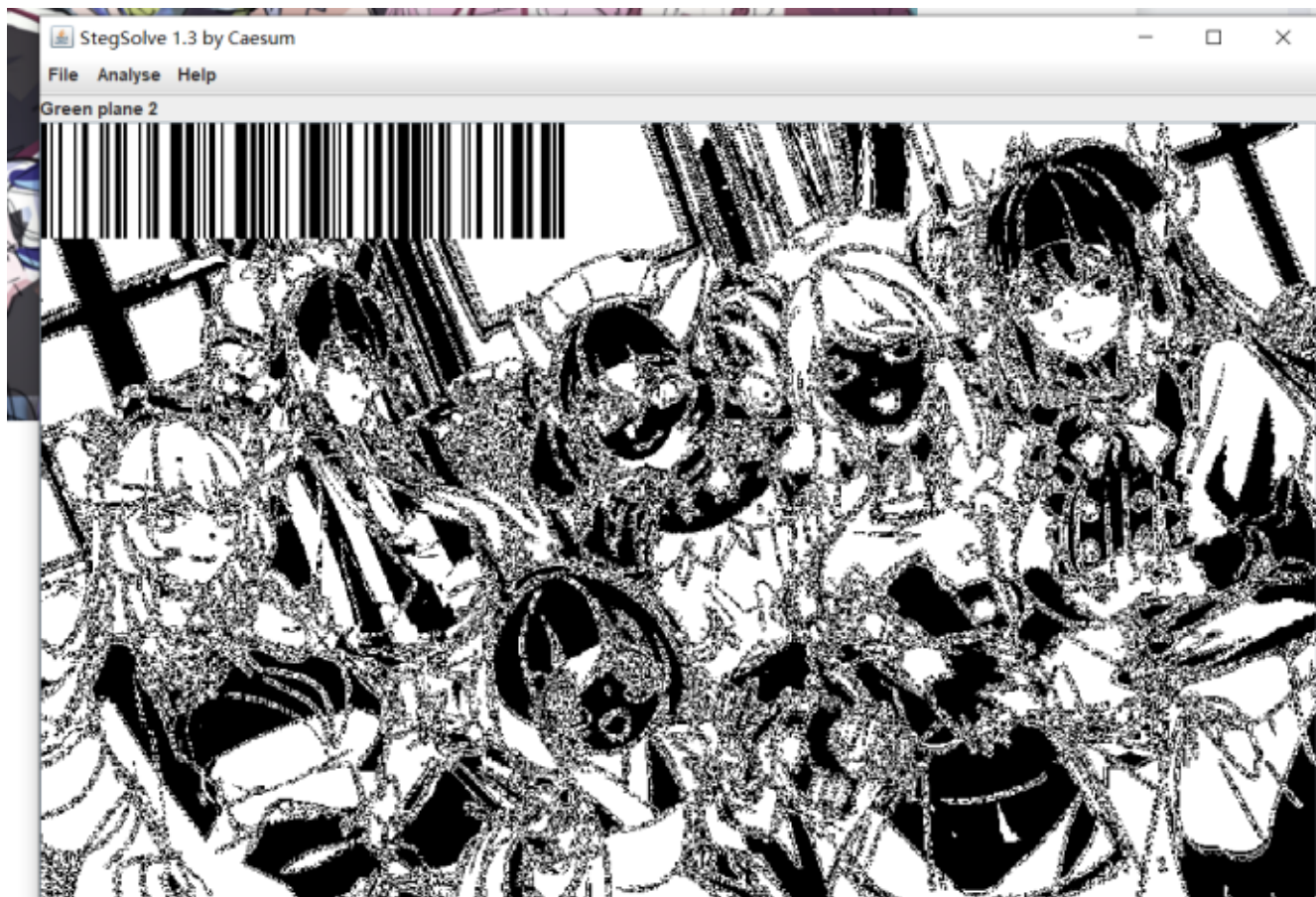
- 1.wireshark打开文件，多点几下，发现一些SMTP有很多字节，选择显示分组字节
- 2.尝试base64解码，看到"PNG"(这里运气好，挑到了第一个)
- 3.又发现蓝色字样"reassembled dat in frame 140"
- 4.到140发现有

```
[Frame: 21, payload: 0-8191 (8192 bytes)]  
[Frame: 23, payload: 8192-16383 (8192 bytes)]  
[Frame: 25, payload: 16384-24575 (8192 bytes)]  
[Frame: 27, payload: 24576-32767 (8192 bytes)]  
[Frame: 29, payload: 32768-40959 (8192 bytes)]  
[Frame: 31, payload: 40960-49151 (8192 bytes)]  
[Frame: 33, payload: 49152-57343 (8192 bytes)]  
[Frame: 35, payload: 57344-65535 (8192 bytes)]  
[Frame: 37, payload: 65536-73727 (8192 bytes)]  
[Frame: 38, payload: 73728-81919 (8192 bytes)]
```

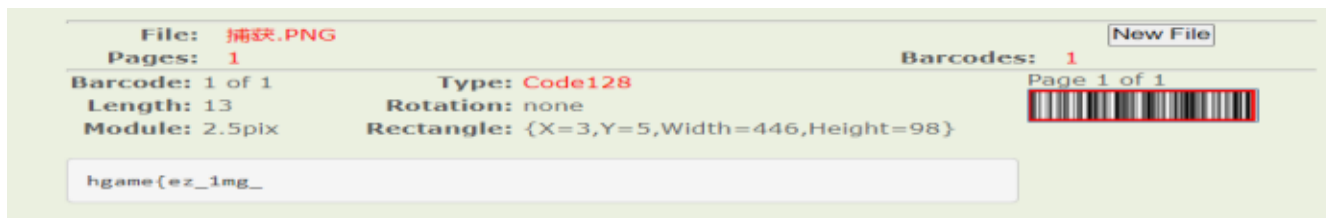
5.跟踪TCP流

```
Content-Transfer-Encoding: base64  
Content-Disposition: attachment;  
filename*=UTF-8''%E6%B6%A9%E5%9B%BE%E2%70%E6%67  
  
iVBORw0KGgoAAAANSUHEugAAA2MAAAIPCAyAAAD+cAacAAEAAAEIEQVR4nOz9eZB1WX7XCX7O  
du99vseSkftamZVZWaWsvJtUqpJKIIEaAUebG1aMZuxXgzNYNPMwHTb9F1b0TYDA3RrpmEG  
EKIQaGGGUQCPIO52kkpb7VVS7bvkvkREZKRhh4dt7dzvznZB+/c++7Z/25h7THRZGZkdVZSPCPC  
33t3e/eec36/7/f3/apP/cGno00AEDkdVv5uoma5KDh99xJLizkmanmh1T+69+0b+72v3fxv  
o27nqNud8/4QIU0m5VL6NhvrE5roaXcC46pkc2tCXTeAosgdc4s5AONxRVU2xBZMYSjyDjc5  
sNA0LdVOTdu0xBgPPi5AKYV11nwxw40ss51oFepPDJYszyiKAlc4iIE61NQ7Jc2kIZpINBAb  
iIByCuUVIQZ8CIQ6AlHOTzuUCLTPH1ahFMQY8crTli1eeUIIhDpACzrTWCwmNyiLCDHSVP5W  
txjtiY28T6GwuWF1NOLMvstYNFTbE8aTmth0++uuv/TfMULrW8qyptqpCTrS1tuU5SWuuvZ9  
FoG2e6GFrcKOG5slw+MKlaXrmT4fQ2Bnc4MQG2KM8lrargoGZwuKxSUW11blu0vHT+Q9d65a  
sKrft6GDSydWufdtjXNjMP/9Hng/K0LtoFvC16gmY1SmUUBbK65u1WxWY6JXrCwWmFyxOa7Z  
aRt8I2cdQ8C3DU1V4ictQSuCNYDgtoa15Rgn1hZZKDKUgjk2bG5WjMuGVu9zvLuP+yy8b3Pj  
evdzvdvdb/u7Y/d2rnUcx72+19rumzSs0SwwGUTFhlGwMy5Z39hke1zivZ95bzcPoTTGwVvY  
YSyF9MwqBbkzrC0WZNawNa4YVw1FnrO2ssIoz9M+DasrK6ytrJISd8gjlX3EGCmrkvwrV9na  
3sbHSI1B8WTCZqTEwc1i4agbz1bZ0viI0hoV101TU453qKuSGEK/Za0U1mw40QixFyhjuUQU  
kdwZlkcZ1ihGBJYWRjz42JNorffOJQdFLOM3RrG4kHNidQljNJeubHx0hZNG9DaQgQfIT57  
fnviQ4AYUQqster5TjHkyTKhtQatNUop1FL9MXnv8V7mDKXox9S29bRTIMAImRqtNcZonLMS  
LeasLucsJbXgp/c3nq3tkitXJ5R15zjk+TpnOLk645zprFLcslVcfHSNpbfFHSySZNKykK  
54sXqL1NG/ZueJ/QwrG81HPn6UWwF3N88EYqhnH6GVee1gemk+3gc0pROMvKUSHyosNZJUr3  
2XWF6v93C0Y88J9v1jjoweZXFncRSile00yL7xyjjvvugv/2nkee/wJ8iw72jgwZ7sfe/1V  
/vnnvsKfevud/MV3vO26tjeMmfXUIcOHYPZOzeX1MV57dXp2jh4KcJlhdSlnaSGtd2rPeNIw  
njQ0recGneahQmuF0YrMGYrcsrIQsb5QkwcGrQ93jbsKEGVEnXkhhkgTPXWTBuy3UCilsGjy  
zGJTWqU6KGxuUY5BURJbqqAVkp+n6UJO0RCiH3CoYNMPPK5a0eMkagDMYRp0qIgeokxElTA  
qST00IXRBh1kYGRQxt4//Mpu/1/6d5QbVg3wMfMvpeS4M400acJ10wVJjNPzU8jEopwe2QdE  
QoxUoaWsgnlAlMGYg6+DUqC9xksZrCFiFXw5BzmCVKAJvTnkeHICoM1ZpokpdCSwWXXZ3Lkr  
ak9QnibuEn90L6iCCYq6CjPXWGeKcbVFVY5RQV/z0OfsEWms1601fkCMgMkieWfW0RBNpMHL  
tTQ6PcgxZd6SsGuV7tm0aA05H+umpw48wc12rda4wmBytfvXvh2343UJrWSSMlphjaIHUwe  
YY3ec0+mHJ3SeIIEIAV74VgXQqRpJymzVp6DTvW0TbeYV4QQAjQgtp1THLTGKKWwxPK5DGPM  
9HfWYowmpLHfGo3RaSyMEZRCa4M2Z1pq51cVIA4Tg1b4lXpDW0vL0tGEAH4ko6q1oJhPG25uz  
2zhUpCsoxow0dTd0ilBQUM2emx4nqkysZQ+g/F9I1765797P7+nQ//SZnXp05qPt9JLJN7wPe  
p3mze7+wBE9rdftCqPHGQ4U1SpK/fQa5iCwCmybg20C6pY4UIUR8svs01hit0329Z+k03W+M  
tCHQth7v4/UtDrvpW83047dwXNm/vkUSMYn9r/n1LvqVURTA3/7pn+ev/9C/4Ie/+BT/+t/+  
DBfPn7/u7T77/Av80//l31FkGb/3zAucfFXVG5I8dts/amgFz5na28yZAw4bEfA+sDNpWN8o
```

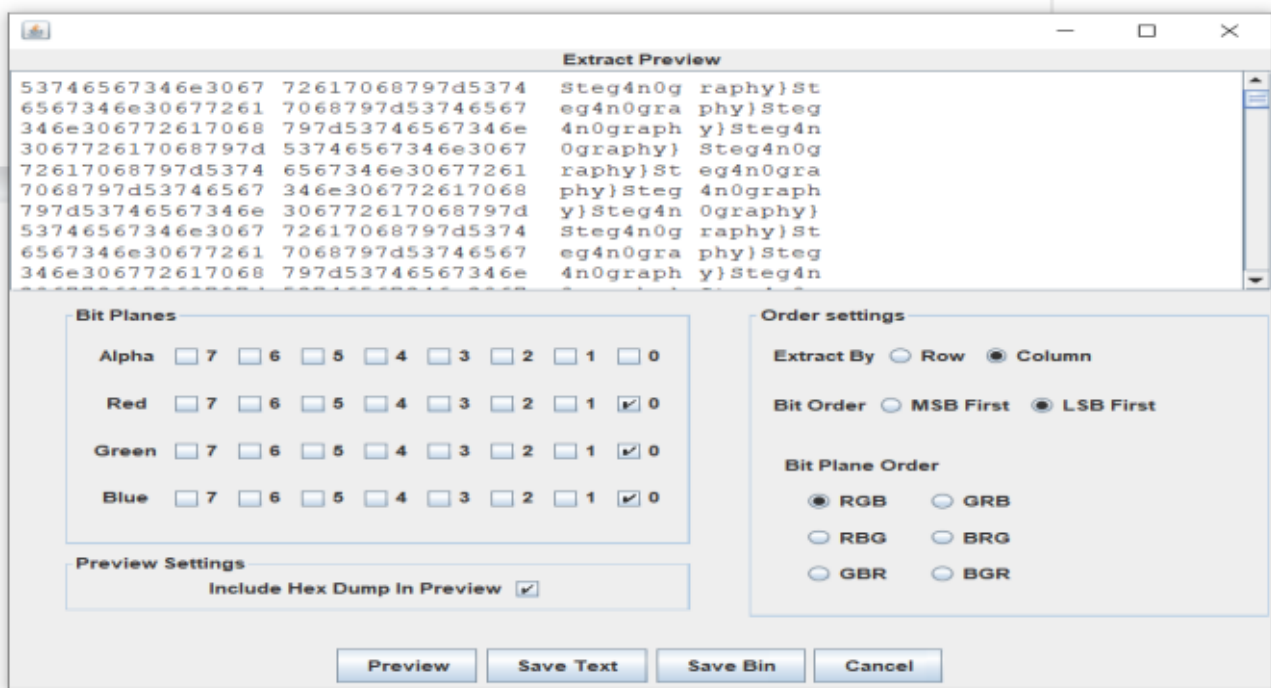
- 6.仅将加密文本保存，用Notepad++打开，并Base64解密



9.条形码扫一扫(这一刻我是很绝望的)



10.图片LSB隐写看一下,发现另一半flag(右边在循环的内容)

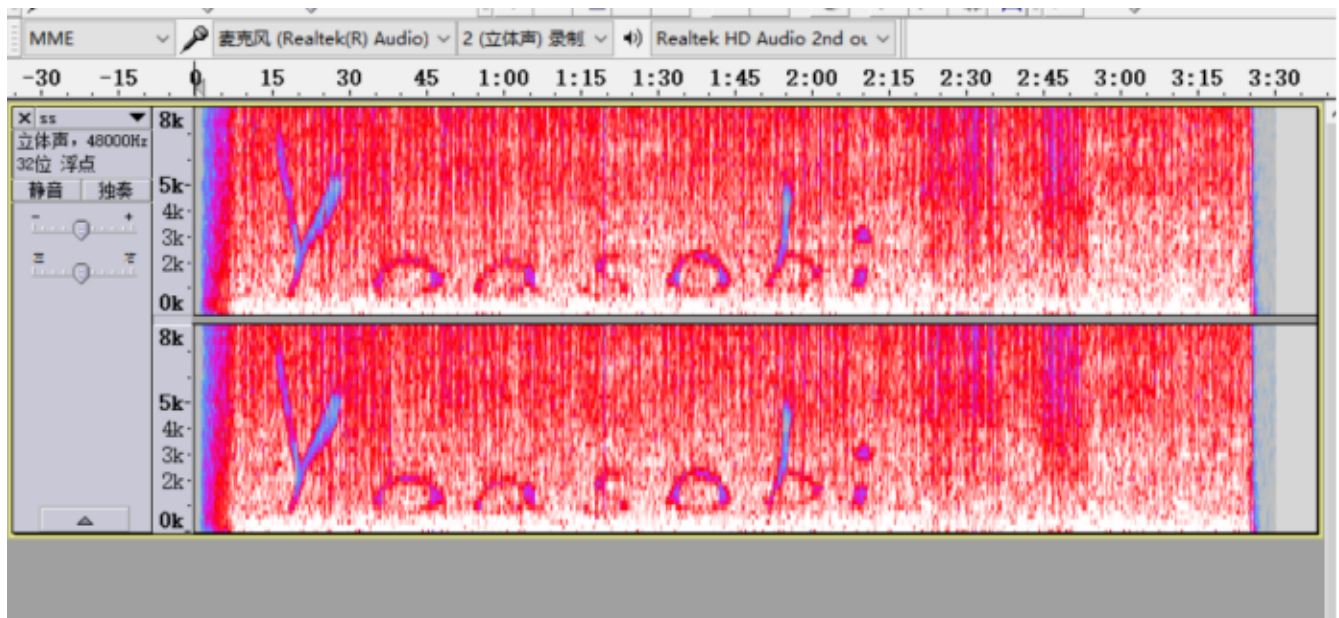


群青(其实是幽灵东京)

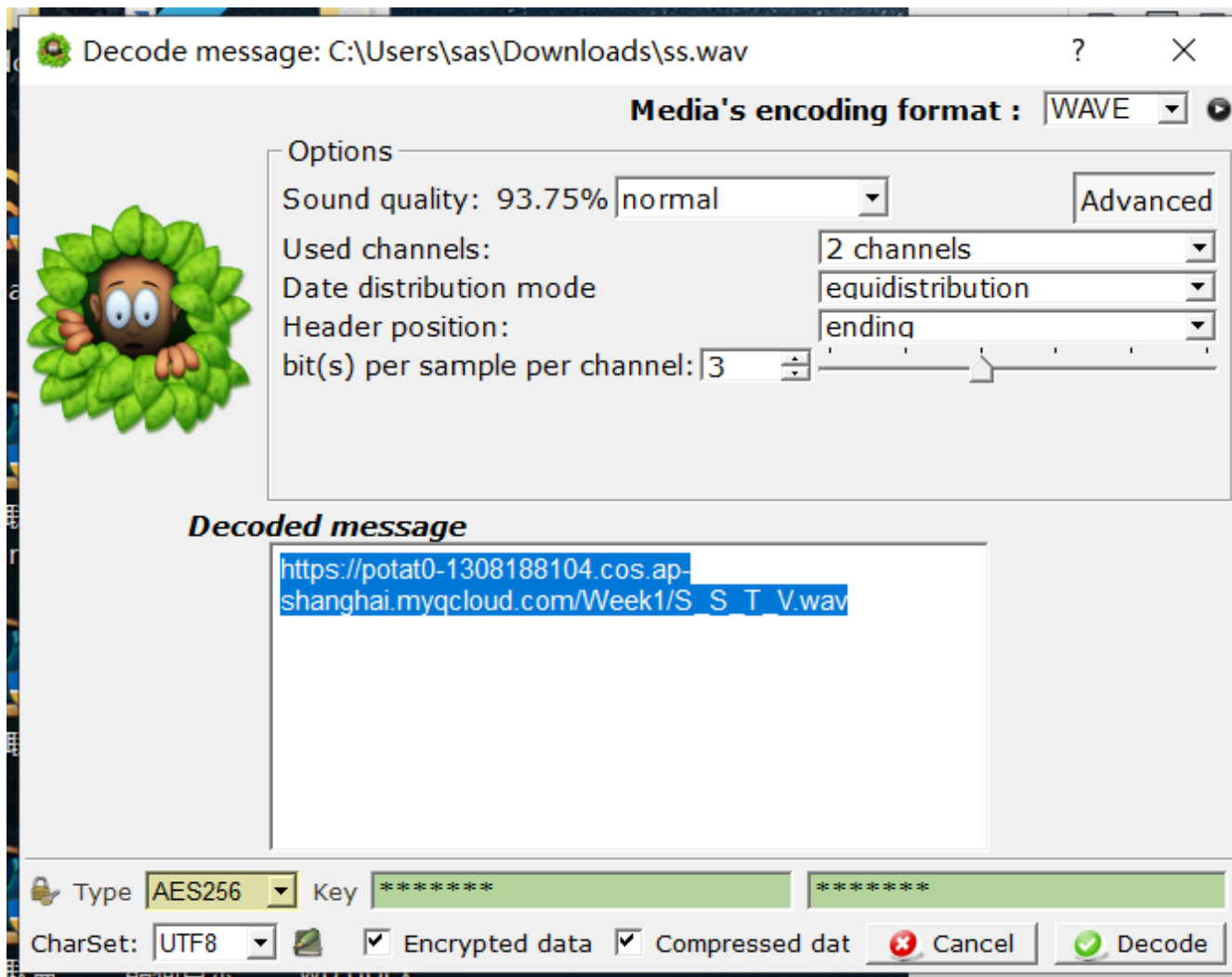
1.winhex打开文件，有提示(silenteye)

77 68 79 20 6E 6F 74 20	74 72 79 20 74 72 79 20	why not try try
53 69 6C 65 6E 74 45 79	65 00 69 64 33 20 2E 00	SilentEye id3 .
00 00 49 44 33 03 00 00	00 00 00 24 54 50 45 31	ID3 \$TPE1
00 00 00 1A 00 00 00 77	68 79 20 6E 6F 74 20 74	why not t
72 79 20 74 72 79 20 53	69 6C 65 6E 74 45 79 65	ry try SilentEye

2.audacity打开文件，频谱图查看（所谓多感官感受）有两个Yoasobi



3.silenteye打开文件解密 key是, Yoasobi Yoasobi



4.进入网址下载第二个S_S_T_V.wav文件，听一下发现是一串滴滴答答的声音

5.(很久之后) 尝试百度sstv，发现"声音传图"!!!! 下载robot36，放音频得到图片



6.扫码得flag

Web

easy_auth

The screenshot displays the Burp Suite Professional v1.7.32 interface. The top menu bar includes "Burp Suite Professional v1.7.32 - Temporary Project - licensed to sas". Below it is a toolbar with tabs: Target, Proxy, Spider, Scanner, Intruder, Repeater, Sequencer, Decoder, Comparer, Extender, Project options, User options, and Alerts. The "Repeater" tab is active. A target URL "http://whatadminisdoingwhat.mjclouds.com" is entered in the address bar. The main workspace is divided into two panes. The left pane, titled "Request", shows the raw HTTP request details: POST /v1/user/login HTTP/1.1, Host: whatadminisdoingwhat.mjclouds.com, User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:96.0) Gecko/20100101 Firefox/96.0, Accept: */*, Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2, Accept-Encoding: gzip, deflate, Referer: http://adminisdoingwhat.mjclouds.com/, content-type: application/json, Origin: http://adminisdoingwhat.mjclouds.com, Content-Length: 38, Connection: close. The body of the request is {"user_name":"","ssaa","passwd":"123456"}. The right pane, titled "Response", shows the raw HTTP response details: HTTP/1.1 200 OK, Server: nginx, Date: Wed, 26 Jan 2022 15:08:32 GMT, Content-Type: application/json; charset=utf-8, Content-Length: 260, Connection: close, Access-Control-Allow-Credentials: true, Access-Control-Allow-Headers: Origin, X-Requested-With, Content-Type, Accept, Authorization, Cookie, token, Access-Control-Allow-Methods: POST, GET, OPTIONS, PUT, DELETE, UPDATE, Access-Control-Allow-Origin: http://adminisdoingwhat.mjclouds.com, Access-Control-Expose-Headers: Content-Length, Access-Control-Allow-Origin, Access-Control-Allow-Headers, Cache-Control, Content-Language, Content-Type, Cache-Control: no-cache. The body of the response is {"code":2000,"message":"success","count":0,"data":{"token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJRRCi6MTU4MiwiaWF0IjE6NjYwOTQyMzI6P5wWwZmZAsEsAAUzLXks"}]. A red circle highlights the "token" value in the response body.

```

HEADER: ALGORITHM & TOKEN TYPE

{
  "alg": "HS256",
  "typ": "JWT"
}

PAYLOAD: DATA

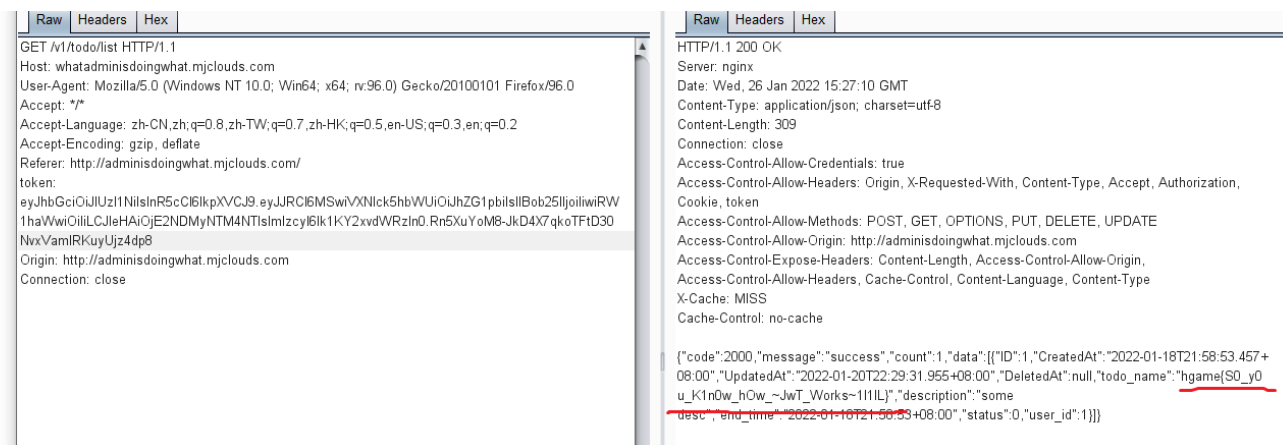
{
  "ID": 1582,
  "UserName": "ssaa",
  "Phone": "",
  "Email": "",
  "exp": 1643252912,
  "iss": "MJclouds"
}

VERIFY SIGNATURE

HMACSHA256(
  base64UrlEncode(header) + "." +

```

4.登录后刷新并抓包，修改token,得到flag



Tetris plus

- 1.游戏得到3000分，出现弹窗"flag 被藏起来了"
- 2.F12打开开发者工具，发现checking.js
- 3.发现base64加密字符串（无用，解出来是弹窗内容）；发现被注释的"[+]++"字符串
- 4.百度，发现JSFuck加密，解密得flag

蛛蛛

- 1.打开开发者工具，发现很简洁，只有一个按钮有链接
- 2.根据代码点下一关，101关时提示"找到了"。开发者工具，网络，响应头找到flag

REVERSE

easyasm

- 1.ida打开文件，发现操作1（16位数据，前8位后8位交换，^17h）
- 2.dseg发现hgame{fill_in_your}（没什么用的样子）；seg001发现一串数据，前几位是hgame执行操作1
- 3.将seg001数据对操作1反向解密，得flag

Flag Checker

- 1.安装，打开，界面有一个文本框，一个click按钮
- 2.ida打开.apk文件看看（我看不懂）
- 3.AndroidKiller打开文件看看(看不懂*2)
- 4.flagchecker.apk解压缩，得classes.dex
- 5.用dex2jar把classes.dex转成classes-dex2jar.jar
- 6.用jd-gui打开classes-dex2jar.jar,看到java代码

```

public class MainActivity extends AppCompatActivity {
    public static byte[] encrypt(String paramString1, String paramString2) throws Exception {
        SecretKeySpec secretKeySpec = new SecretKeySpec(paramString2.getBytes(), 0, paramString2.length(), "RC4");
        Cipher cipher = Cipher.getInstance("RC4");
        cipher.init(1, secretKeySpec);
        return cipher.doFinal(paramString1.getBytes());
    }

    protected void onCreate(Bundle paramBundle) {
        super.onCreate(paramBundle);
        setContentView(2131296284);
        ((Button)findViewById(2131165218)).setOnClickListener(new View.OnClickListener() {
            public void onClick(View param1View) {
                String str = ((EditText)findViewById(2131165238)).getText().toString();
                byte[] arrayOfByte = new byte[0];
                try {
                    byte[] arrayOfByte1 = MainActivity.encrypt(str, "carol");
                    arrayOfByte = arrayOfByte1;
                } catch (Exception exception) {
                    exception.printStackTrace();
                }
                if (Base64.encodeToString(arrayOfByte, 0).replace("\n", "").equals("mg6CITV6GEaFDTYnObFmENOAVjKcQmGncF90WhqvCFyhhsyqq1s=")) {
                    Toast.makeText((Context)MainActivity.this, "Congratulations!!!", 1).show();
                } else {
                    Toast.makeText((Context)MainActivity.this, "Fail, try again.", 1).show();
                }
            }
        });
    }
}

```

7.可以猜测, flag经过以"carol"为密钥的"RC4"加密, 又经过base64加密, 得到密文

8.密文"mg6CITV6GEaFDTYnObFmENOAVjKcQmGncF90WhqvCFyhhsyqq1s="解密得flag