Happy2023!

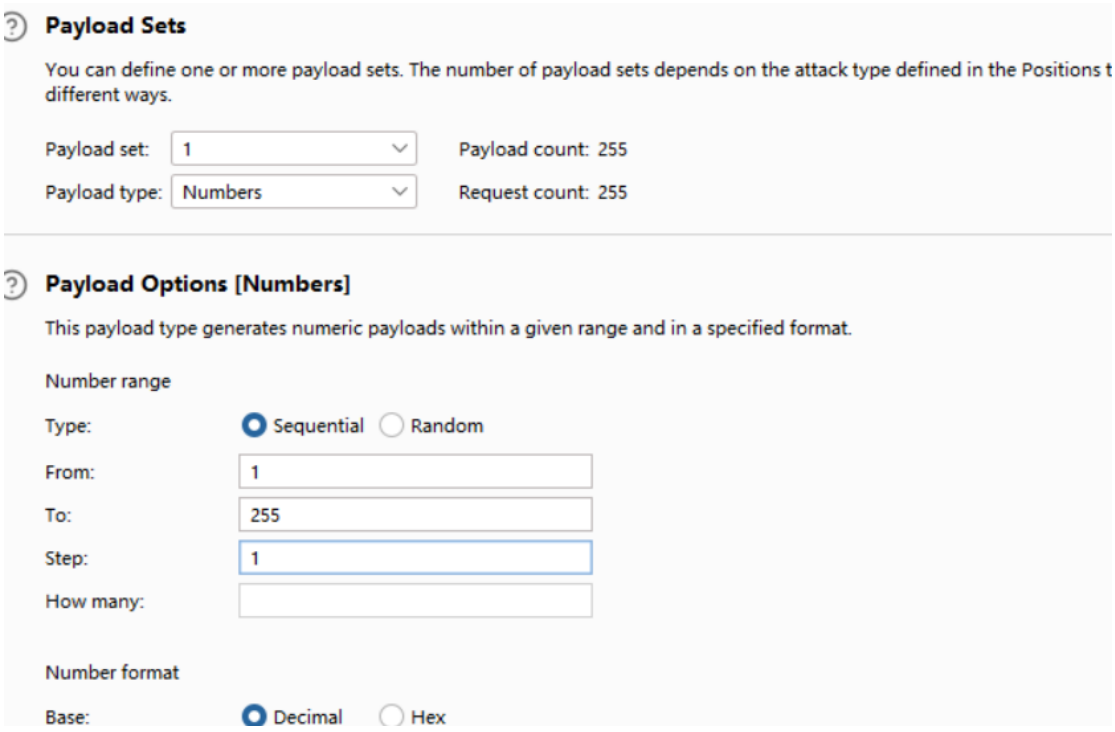
按照第三阶段提示进入目标网页，然后发现需要登录，但是无法输入，直接 f12 修改 disable，然后输入自己的 token 进行登录。进入如下界面：


根据规则要求发现是前 8 位不同才算是一个新用户，那么直接助力抓包看看。

```
POST /invite/bfd0639a-bbac-415a-a95e-d6f8bf8799f2 HTTP/1.1
Host: ilrxbzij.weilei.studio
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/109.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
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
Content-Type: application/x-www-form-urlencoded
Content-Length: 16
Origin: http://ilrxbzij.weilei.studio
Connection: close
Referer: http://ilrxbzij.weilei.studio/invite/bfd0639a-bbac-415a-a95e-d6f8bf8799f2
Cookie: _clck=1xpfm9x|1|f8n|0; cf_clearance=5oqDfdaBgea6_DwF7F01X7hh5mqG1EpieuPFvVotiB8-1674901171-0-250; session=
eyJ0b2tlbiI6Ijk20klFVUNJUUR3V1dBbmlKUGVGT25ueWE2MVlvSnZtQktmWnRKbjBBRDEyRlZnamY4UUdnSWdJUTlkbWRlMmJwbjc3b0JjLXBYVGttcON4ak9CalFnW
HBmR3JKW1B2RDNNPSJ9.Y9T6rQ.rlNk7lR7kQszmgA8yoelb69KJqc
Upgrade-Insecure-Requests: 1

ip=1.142.153.83
```

抓包内容如图，直接爆破去修改前 8 个 bit 从 1 到 255 即可。设置 payload




flag{BytesSec祝大家新年财源广进}，支付宝红包口令：BytesSec祝大家新年财源广进，请确保您提交flag后领取红包以便统计人数，请勿泄露口令！

签到

Base100 emoji 编码

欢迎来到躺平杯！这是flag1: flag{Happy_NEW_YEAR_2023}

二维的码

https://products.aspose.app/barcode/zh-hans/recognize#识别各类二维码的网址
打开给的二维码发现什么都不像但是又有点规整，直接 stegsolve 黑白反转


根据提示汉明码进行识别

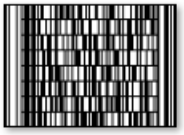
Good !Flag1: flag{domestic_2D_code} Go /media/boarding_pass.zip

然后去 get 到第二个文件发现有密码，先拼接好第二个二维码缺少三个定位符，直接手动 ps

然后进行扫描。

flag2:flag{L00k_l1k3_bo4rd1ng_p4ss}
flag3:/media/enterpoint.gif

Go /media/c0d3f14g3.zip


part1.gif


part2.gif


pass.gif

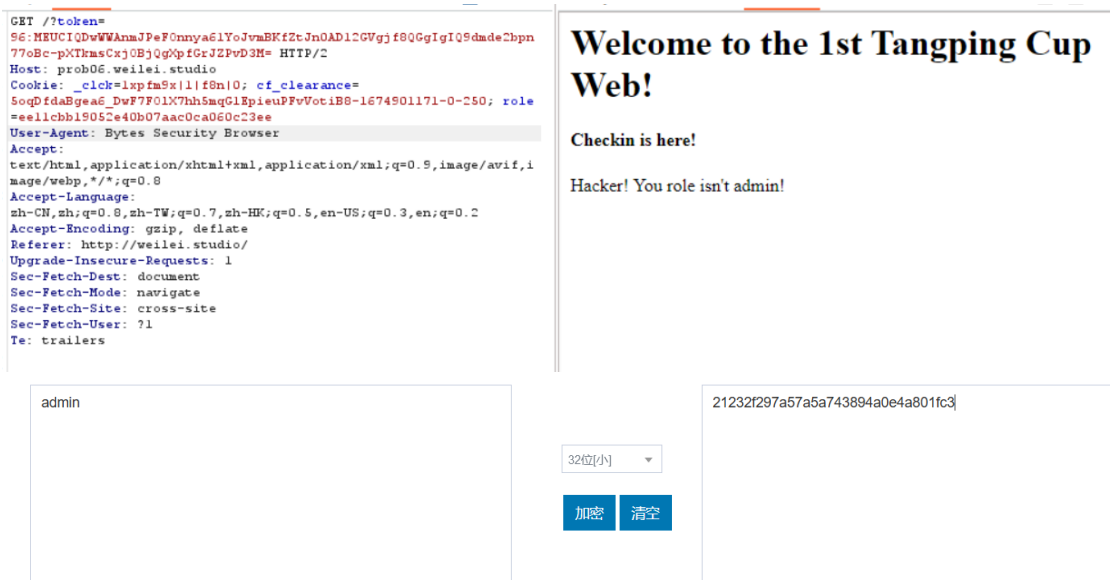
分别进行扫描，有一个叫 pass 应该是加密算法，根据提示应该是 rabbit 加密。

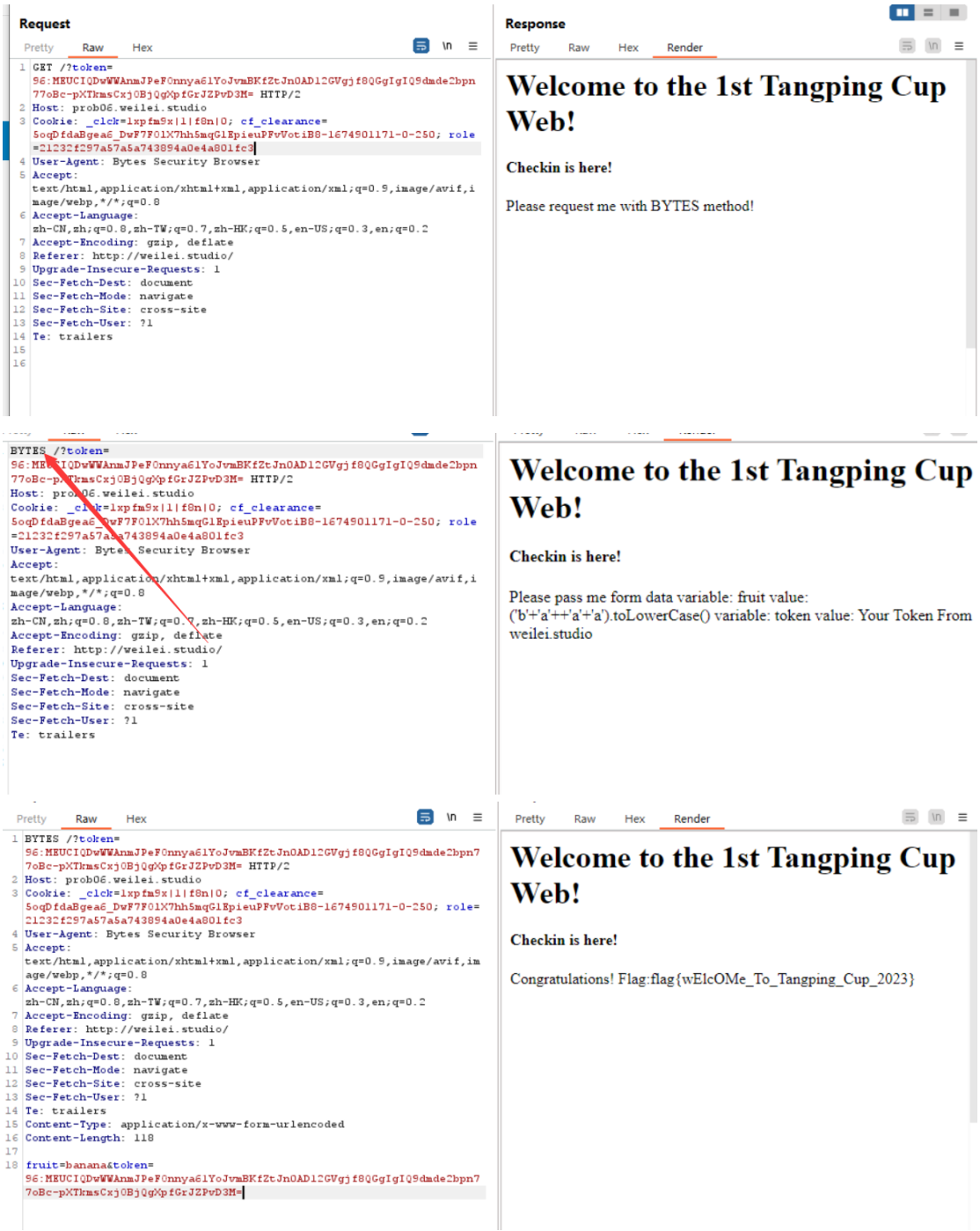

HTTP 百解

F12

```
▶ <p> ··· </p>
  <!--Flag1: flag{WoW_W3B_so_3asy!I_1iKe_Tangp1ng_CUp!}. Remember take your token to get flag2-->
</bodv>
```

Please request me securely with Bytes Security Browser !






参考将 get 改为 post 所需要添加的一些东西即可

躺平问答 • Beta 版

点击 submit 发现 url 传入了一个参数，将参数 base64 解码

```
OTY6TUVVQ0lRRHdXV0FubUpQZUZPbm55YTYxWW9Kdm1CS2ZadEpuMEFEMTJHVmdqZjhRR2dJZ0lROWRtZGUyYnBuNzd
hUa21zQ3hqT0JqUWdYcGZHckpaUHZEM009fDA=
```

Output

start: 101 time: 2ms
end: 101 length: 101
length: 0 lines: 1

```
96:MEUCIQDwWWAnmJPeFOnnya61YoJvmBKfZtJn0AD12GVgjf8QGgIgIQ9dmde2bpn77oBc-
pXTkmsCxjOBjQgXpfGrJZPvD3M=|0
```

末尾将 0 改为 114514 然后再编码传入参数。

flag{woW_tAngping_wEnda_is_soeaSy}

WearMe 核心机房 准入验证

第一部分根据万能密码的提示将万能密码填入用户 admin’ or ’1’=’1

登陆成功!Flag1: flag{wo0ps_H4cker_IS_H3re}

第二题不会

我抄，盒！

```
6D     Apple iPad m
48 ini   ü€  '    H
67     flag{F1nd1ng
30 _FakE_eDiting_S0
30 ft_w4re}  2077:0
1F 1:23 15:39:17
05 ,š         œ,
01       ¤^"
```

winhex 打开照片

第二部分直接搜索 exif 查看器然后就可以了

口算大整数

# 一、低加密指数攻击

低加密指数攻击的原理此处不再赘述，仅列举题型和exp。

# 二、低加密指数广播攻击

识别：n非常大,e一般很小。

rsa 常见的攻击
代码如下：

```
from gmpy2 import iroot
import libnum
e = 0xa
n =
0x52a0f8cd157cc045ceeb12d310d736cab14024c9e339ad7468daec43198bd6842b9bd
233b1174e47b483bfe0b3025a890bfdd767abcb87721661599811b745df220ba28d2d07
a515ef411a822fe463fa7a32f2d935c21b0934d26346a808860d3dd70378abcc246706e
9b9f70f75af4dc67b8a69fd6b6f165ef1bf0d75293b94b8517b34c7bdb661d45910298e
3b0066dac39a8557e32cc482c7f8c981d993beaa4c3fb6dc338606973834e1058d60d11
d2fc73757d355e139d5404d6ec6e9e603d75962d5c3888ab0c25b8a7daaa3bf3a896cc0
b2f4314fdf4b4355a263731ef74207e0e199c0413b5be6b4c905ca6cf351445abb7e2ee
3a839
c =
0x6e33c752c8c1c155b934bcfa66ceffa1ef39aa914860a7f33911497760f9c0dded2ce
31b552d14b63cda7c0bbb60839de8010c6b283c44ec7afbd70fbf981d774e6ca986a87e
e0e40911faa7b14ec36818f47e33a5de0c453ade309e6710c0b9faec9b1cd9bfc39c992
76fe7bab3dd52d872a6b7f73010cab259bca7bdb6c44b5ba3a1c04299bdfdd0403d9f5a
901385ea4f6124dd639d46f121fb2ec2cb17227a66abbe172cefa28feb1cdd1cd773149
58048a9cc0f0d217a744c369121487c8033b52495e44ba8fd06c779a5368ac308034a75
2b9ed52a4c5416f86fa1702f5120ec06b5c901a89f0d528cbd78542bd8812e7aa8ac399
a9

k = 0
while 1:
    res = iroot(c+k*n,e)
    #print(res)
    #res =
(mpz(13040004482819713819817340524563023159919305047824600478799740488797710355579494486728991357), True)
    if(res[1] == True):
        print(libnum.n2s(int(res[0]))) #转为字符串
```

```
        print(int(res[0]))
        break
    k=k+1
```

V 我≈50

我应该是非预期做出来的吧。。。
直接本地运行脚本然后运行后的结果多试几次就正确了 QWQ

躺平 flag 提取器

拖入 ida 分析
Shift+f12 直接找到一个

flag{wHat_The_fxxK_1t_1s}

然后发现其中还有一个看起来奇怪的猜测是进行了某种运算

synt{mu6_K6R_JR6_Wv_6f_P5ZvAT}

有一个 rot13 函数估计就是关键了，直接 vscode 打开照抄代码逻辑，如下图：

```
result = ''
flag = 'synt{mu6_K6R_JR6_Wv_6f_P5ZvAT}'

for i in flag:
    if i.isalpha() == True:
        if ord(i.lower()) > 110:
            result += chr(ord(i) - 13)
        else:
            result += chr(ord(i) + 13)
    elif (ord(i) - 48) > 9:
        result += i
    elif ord(i) > 52:
        result += chr(ord(i) - 5)
    else:
        result += chr(ord(i) + 5)
print(result)
```

```
PS D:\kali_pwn_share> py test.py
fl{g{zh1_X1E_WE1_Ji_1s_C0MiNG}
```

躺平 flag 提取器 • 极速版

找一个反编译 apk 的软件，这里我用的是 GDA 比较方便点，不需要提取其他东西。

```
{
    String encryptedFlag1;

    public void FlagActivity1(){
        super();
        this.encryptedFlag1 = "flag{FAST_Cracked}_Just_JiJiJi";
    }
    public boolean isCracked(){
        return false;
    }
    protected void onCreate(Bundle savedInstanceState){
        super.onCreate(savedInstanceState);
        this.setContentView(R.layout.activity_flag1);
        View view = this.findViewById(R.id.flag1Layout);
        TextView textView = this.findViewById(R.id.textViewo0oooo00000o0000);
        TextView textView1 = this.findViewById(R.id.textViewO000000000000000);
        TextView textView2 = this.findViewById(R.id.textView0o0000ooooo0oooo);
        TextView textView3 = this.findViewById(R.id.textView0000000000000000);
        if (this.isCracked()) {
            textView.setText((this.encryptedFlag1).substring(18, 24));
            textView1.setText((this.encryptedFlag1).substring(10, 18));
            textView2.setText((this.encryptedFlag1).substring(0, 10));
            textView3.setText((this.encryptedFlag1).substring(24, 30));
            Snackbar.make(view, "成功获取flag! ", -2).show();
```

直接发现 flag1 查一下 substring 是什么意思，然后直接开始写代码。

```
flag = 'flag{FAST_Cracked}_Just_JiJiJi'
result = ''
result += flag[18:24]
result += flag[10:18]
result += flag[0:10]
result += flag[24:30]
print(result)
```

```
_Just_Cracked}flag{FAST_JiJiJi
```

再手动改一下顺序即可。

第二部分叫安全提取，并且在 flag2 函数中也没发现什么有用的信息，怀疑是被隐藏了，直接使用 winhex 也找不到。发现解压后有个.so 文件比较重要。直接拖入 ida 打开看看。

| Address | Length | Type | String |
|---|---|---|---|
| LOAD:000000... | 00000008 | C | Android |
| LOAD:000000... | 0000000D | C | __cxa_atexit |
| LOAD:000000... | 0000000F | C | __cxa_finalize |
| LOAD:000000... | 00000024 | C | Java_com_zhizi42_ctf_Native_getFlag |
| LOAD:000000... | 0000000E | C | _Z9isCrackedv |
| LOAD:000000... | 0000001E | C | _ZN7_JNIEnv12NewStringUTFEPKc |
| LOAD:000000... | 00000008 | C | libc.so |
| LOAD:000000... | 00000008 | C | libm.so |
| LOAD:000000... | 00000009 | C | libdl.so |
| LOAD:000000... | 0000000B | C | libmain.so |
| .rodata:00000... | 00000027 | C | flag{Get_the_flag_safely_from_SO_file} |
| .rodata:00000... | 00000018 | C | you haven't cracked yet |

Ok 结束！可以再去看看 android 的文件打包过程。