RE Easyasm
使用 ida

```
cmp     si, 1Ch
```

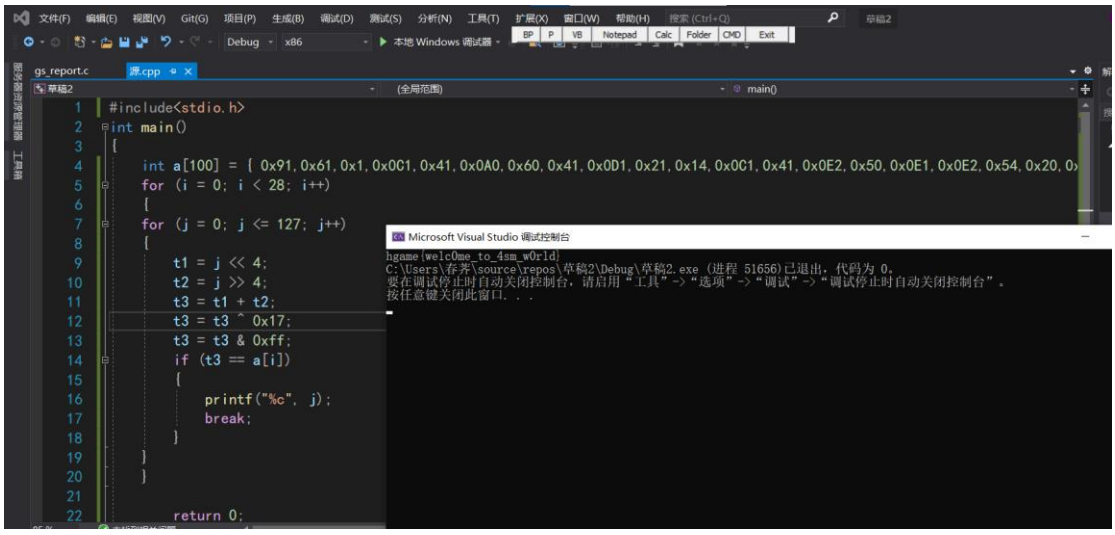
密码长度 28 位

```
seg001:0000                 db  91h
seg001:0001                 db  61h ; a
seg001:0002                 db    1
seg001:0003                 db 0C1h
seg001:0004                 db  41h ; A
seg001:0005                 db 0A0h
seg001:0006                 db  60h ; `
seg001:0007                 db  41h ; A
seg001:0008                 db 0D1h
seg001:0009                 db  21h ; !
seg001:000A                 db  14h
seg001:000B                 db 0C1h
seg001:000C                 db  41h ; A
seg001:000D                 db 0E2h
seg001:000E                 db  50h ; P
seg001:000F                 db 0E1h
seg001:0010                 db 0E2h
seg001:0011                 db  54h ; T
seg001:0012                 db  20h
seg001:0013                 db 0C1h
seg001:0014                 db 0E2h
seg001:0015                 db  60h ; `
seg001:0016                 db  14h
seg001:0017                 db  30h ; 0
seg001:0018                 db 0D1h
seg001:0019                 db  51h ; Q
seg001:001A                 db 0C0h
seg001:001B                 db  17h
```

未解密的 28 位密码 16 进制

```
xor     ax, ax
mov     al, [si]
shl     al, 1
shl     al, 1
shl     al, 1
shl     al, 1
push    ax
xor     ax, ax
mov     al, [si]
shr     al, 1
shr     al, 1
shr     al, 1
shr     al, 1
pop     bx
add     ax, bx
xor     ax, 17h
add     si, 1
cmp     al, es:[si]
jnz     short loc_100DD
```

将每一位输入的数左移四位，
同时将原来输入的数右移四位，
两者相加，再与 0x17 异或，与密文比较
编写 c 语言得到 flag



RE Flagchecker
使用 AndroidKiller 打开 apk

```
invoke-static {p1, v0}, Lcom/example/flagchecker/MainActivity;->encrypt(Ljava/lang/String;Ljava/lang/String;)[B
```

第一处加密
进入 MainActivity 下的 encryt 方法

```
const-string v3, "RC4"

invoke-direct {v0, v1, v2, p1, v3}, Ljavax/crypto/spec/SecretKeySpec;-><init>([BIILjava/lang/String;)V
```

加密方式为 RC4

```
const-string v0, "carol"
```

找到密钥为 carol

```
const-string v0, "mg6CITV6GEaFDTYnObFmENOAVjKcQmGncF90WhqvCFyhhsyqq1s="
```

密文为 mg6CITV6GEaFDTYnObFmENOAVjKcQmGncF90WhqvCFyhhsyqq1s=

解密后得到 hgame{weLC0ME_To-tHE_WORLD_oF-AnDr0|D}

RE Crackme

使用 od 打开
找到加密后的密文
88 34 D9 48 4C 14 0C 03 .
C2 78 EB 52 ED E5 9C ED
E6 ED 1F AE 6D 12 5A BA
AA 84 92 CF E3 F2 E0 65

在 ida 中 

加密函数

```
strcpy(&code, "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=");
k = 0;
offset_old = 0;
offset = 0;
do
{
  t1 = *(int *)((char *)&input + offset);
  t2 = *(int *)((char *)&v14 + offset);
  v15 = 0;                                              // v15并没有用处
  j = 32;
  do
  {
    k += 0x12345678;
    t1 += k ^ (k + t2) ^ (v9 + 16 * t2) ^ (v10 + (t2 >> 5));
    t2 += k ^ (k + t1) ^ (code + 16 * t1) ^ (v8 + (t1 >> 5));
    --j;
  }
  while ( j );
  temp = offset_old;
  k = 0;
  *(int *)((char *)&input + offset_old) = t1;
  *(int *)((char *)&v14 + temp) = t2;
  offset = temp + 8;
  offset_old = offset;
}
while ( offset < 32 );
v11 = xmmword_402180;                                   // 密文
v12 = xmmword_402170;
while ( *((_BYTE *)&input + k) == *((_BYTE *)&v11 + k) )
{
  if ( ++k >= 32 )
  {
    sub_40100C((int)"right!");
    return 0;
  }
}
sub_40100C((int)"wrong!");
return 0;
```

以八位十六进制为单位进行加密

```
j = 32;
do
{
    k += 0x12345678;
    t1 += k ^ (k + t2) ^ (0x4c4b4a49 +16 * t2) ^ (0x504f4e4d +(t2 >> 5));
    t2 += k ^ (k + t1) ^ (0x44434241 +16 * t1) ^ (0x48474645 +(t1 >> 5));
    --j;
} while (j);                                    // 循环32次
k = 0;
```

加密模块 c 语言表示

现在编写解密代码

```c
#include<stdio.h>
int main()
{
    int j;
    unsigned long t1, t2, k;
    t1 = 0x48d93488;
    t2 = 0x030c144c;
    k = 0x468acf00;
    j = 32;
    do
    {
        t2 -= k ^ (k + t1) ^ (0x44434241 + 16 * t1) ^ (0x48474645 + (t1 >> 5));
        t1 -= k ^ (k + t2) ^ (0x4c4b4a49 + 16 * t2) ^ (0x504f4e4d + (t2 >> 5));
        k -= 0x12345678;
        --j;
    } while (j);
    printf("%x%x", t1, t2);
}
```

将密文输入后依次输入为

6d61676834487b65 magh 4H{e

5f79707034633476 _ypp 4c4v

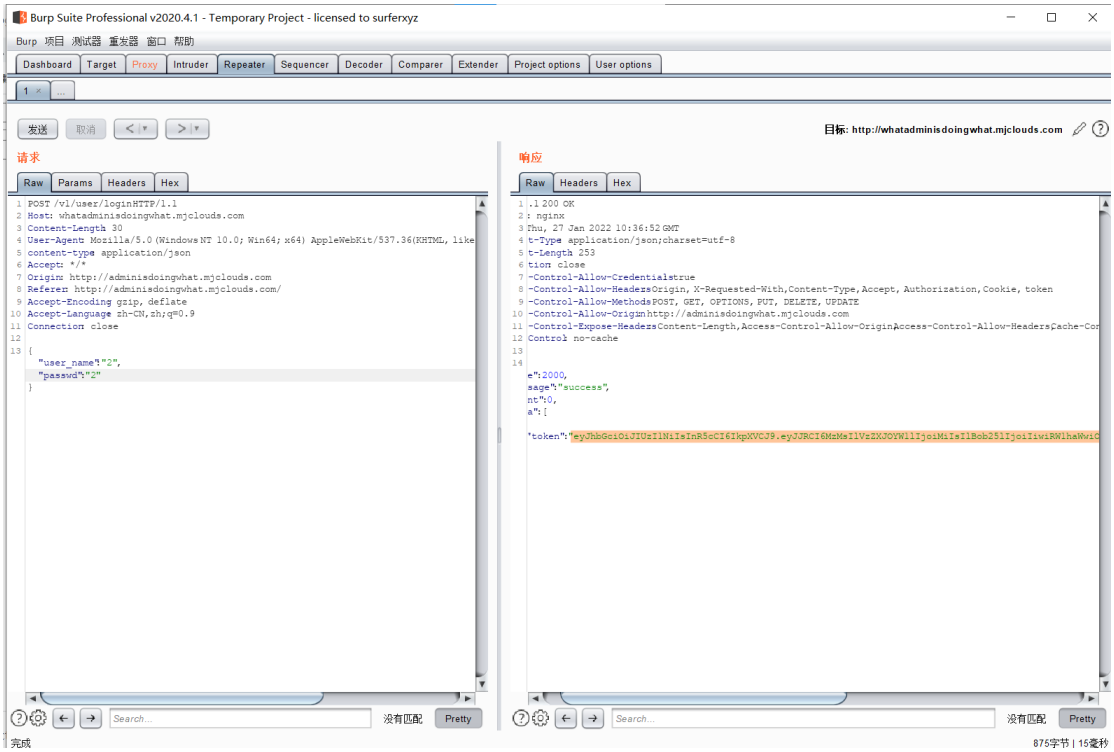
6e30697400007d21 n0it }!

以四位为单位倒序拼接后得到 flag

hgame{H4ppy_v4c4ti0n!}

WEB easy_auth

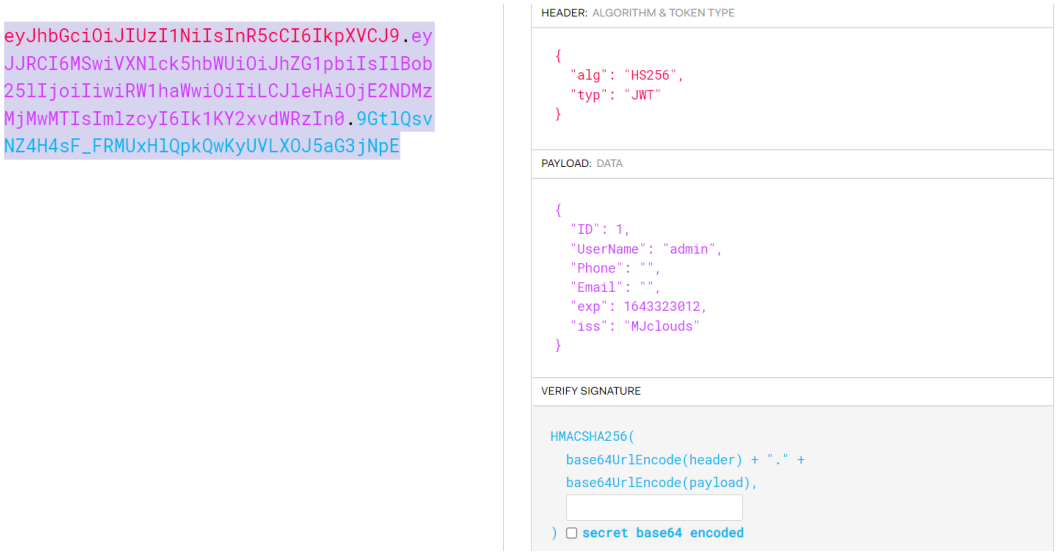
使用 burpsuit 抓包



找到 token

"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJRCI6MzMsIlVzZXJOYW1lIjoiMiIsIlBob25lIjoiIiwiRW1haWwiOiIiLCJleHAiOjE2NDMzMjMwMTIsImlzcyI6Ik1KY2xvdWRzIn0.1ioBKmvstFCf4ABJV6

GhrAsHlxwAkz6WVI7n9X0AILU"
打开 jwf.io
修改 id 为 1，username 为 admin



得到 tokeneyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJRCI6MSwiVXNlck5hbWUiOiJhZG1pbiIsIlBob25lIjoiIiwiRW1haWwiOiIiLCJleHAiOjE2NDMzMjMwMTIsImlzcyI6Ik1KY2xvdWRzIn0.9GtlQsvNZ4H4sF_FRMUxHlQpkQwKyUVLXOJ5aG3jNpE
修改 login.js 代码

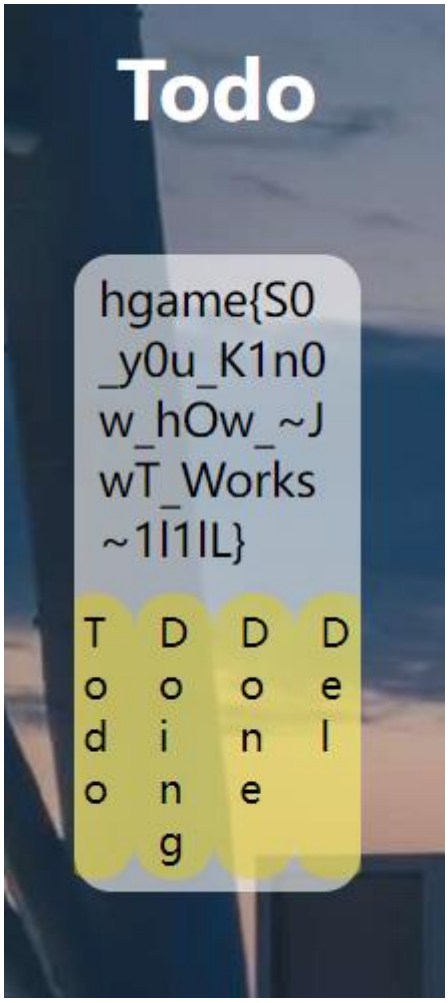
```
function Login() {
    postData("/v1/user/login", {
        "user_name": uname.value,
        "passwd": passwd.value
    }).then((data) => {
        console.log(data);
        if (data.code !== 2000) {
            alert(data.message)
        } else {
            localStorage.setItem("token", data.data[0].token)
            console.log(localStorage)
            window.location.href = "/"
        }
    })
}
```

为

```
function Login() {
    postData("/v1/user/login", {
        "user_name": uname.value,
        "passwd": passwd.value
    }).then((data) => {
        console.log(data);
        data.data[0].token="eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJRCI6MSwiVXNlck5hbWU
            localStorage.setItem("token", data.data[0].token)
            console.log(localStorage)
            window.location.href = "/"

    })
}
```

保存后任意输入注册的用户名和密码



得到 flag

hgame{S0_y0u_K1n0w_hOw_~JwT_Works~1l1lL}