Catalina

Chạy thử chương trình

```
┌──(kali㉿ DESKTOP-6TCCJEL)-[/mnt/c/Users/BILL/Desktop/Release_3/Catalina]
└─$ ./crackme aaaa
Invalid flag, try again
```

Phân tích với IDA

```
v12 = "JTQSRyZKSB05Dh9JgH6fQJIVjJ04UpA7ezxMIHcvpX6X70NJHW4xlxSHHMuLDjCJbzl9ITfgeLbTDLExZENyYrAzn7ehjAMuZf1siTB4HBLgyJ"
      "gpK38LHCq4UvpgqOxeoh72AVgDOYS8HU9xg";
v11[0] = 4;
v11[1] = 4;
v11[2] = 5;
v11[3] = 4;
v11[4] = 2;
v11[5] = 4;
v11[6] = 3;
v11[7] = 4;
v11[8] = 2;
v11[9] = 4;
v11[10] = 6;
v11[11] = 2;
v11[12] = 4;
v11[13] = 6;
v11[14] = 2;
v11[15] = 5;
v11[16] = 5;
v11[17] = 2;
v11[18] = 3;
v11[19] = 3;
v11[20] = 5;
v11[21] = 4;
v11[22] = 2;
v11[23] = 3;
v11[24] = 4;
v11[25] = 2;
v11[26] = 2;
v11[27] = 3;
v11[28] = 3;
v11[29] = 2;
```

```
54    v9[1] = 0LL;
55    v9[2] = 0LL;
56    v9[3] = 0LL;
57    v10 = 0;
58    v15 = 0;
59    for ( i = 0; i <= 31; ++i )
60    {
61      *((_BYTE *)v9 + i) = v12[v15];
62      v15 += v11[i] + 1;
63    }
64    sub_1482(v9, v8, 32LL);
65    v15 = 0;
66    v13 = 0;
67    while ( v15 <= 23 )
68    {
69      v8[v15] ^= 0x41u;
70      v4 = (unsigned __int8)v8[v15];
71      v5 = a2[1];
72      v6 = strlen(v5);
73      if ( v6 >= v15 )
74        v7 = v15;
75      else
76        v7 = strlen(a2[1]);
77      v13 += v4 == v5[v7];
78      ++v15;
79    }
80    if ( v13 == 24 )
81      puts("Congratulations !! you solved the first challenge.");
82    else
83      puts("Invalid flag, try again");
84    result = 0LL;
85  }
```

Chúng ta chỉ cần để ý từ dòng 67 đến 83. Chúng ta cần v13 == 24.

Vòng lặp while chạy 24 lần, mỗi lần lặp lấy (kí tự của chuỗi v8 ^ 0x41) và so sánh với kí tự thuộc chuỗi pass chúng ta nhập vào (a2[1]). Chúng ta cần điều kiện này đúng cả 24 lần lặp để câu lệnh (v13 += (v4 == v5[v7])) đạt được v13 == 24.

Chuỗi v8 được tạo rất phức tạp, tuy nhiên nó luôn cố định, vì vậy ta chỉ cần đặt breakpoint để xem chuỗi của nó.

```
D0 db  27h ; '
D1 db  2Dh ; -
D2 db  20h
D3 db  26h ; &
D4 db  3Ah ; :
D5 db  73h ; s
D6 db  71h ; q
D7 db  73h ; s
D8 db  71h ; q
D9 db  1Eh
DA db  32h ; 2
DB db  20h
DC db  2Fh ; /
DD db  20h
DE db  1Eh
DF db  32h ; 2
E0 db  20h
E1 db  72h ; r
```

Sử dụng python để xor các giá trị và tìm ra chuỗi cần nhập

```
>>> a = [0x27, 0x2d, 0x20, 0x26, 0x3a, 0x73, 0x71, 0x73, 0x71, 0x1e, 0x32, 0x20, 0x2f, 0x20, 0x1e, 0x32, 0x20, 0x72, 0x2
8, 0x25, 0x20, 0x7b, 0x68, 0x3c]
>>> print(''.join([chr(i ^ 0x41) for i in a]))
flag{2020_sana_sa3ida:)}
```

flag{2020_sana_sa3ida:)}

```
┌──(kali㉿ DESKTOP-6TCCJEL)-[/mnt/c/Users/BILL/Desktop/Release_3/Catalina]
└─$ ./crackme "flag{2020_sana_sa3ida:)}"
Congratulations !! you solved the first challenge.
```