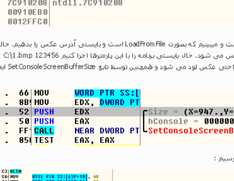
در چهارمین دوره مسابقات دانشگاه شریف بخش reverse دومین challenge یک فایل PE قرار داده شده که اگر با RDG چک کنیم می بینیم فایل A.Net است و هیچ لایه محافظتی بروی آن وجود ندارد و همچنین از crypto خاصی استفاده نمی کند.

این پنجم Green نام دارد که در ادامه به مبحوث بودن نام با توجه حل آن مشخص خواهد شد.طبرقرنک ممکن است برای بدست آوردن فلگ روند خاصی برای خودش داشته باشد ولی ما به روش خود حل خواهیم نمود.

این فایل یک برنامه CLI است پس لینک بیشتر عاقلی اجرا میکنیم متوجه می شویم که بو پارامتر ورودی نیاز دارد. حال بیاییم با پارامتر فاک اجرا میکنیم: HamID 123456

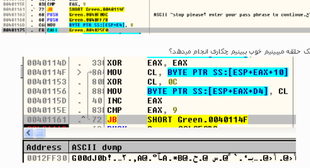


اولین سرنخ نمیدونم. اولین سرنخ میدونم. اولین سرنخ اجرا میکنیم و به صورت زیر اینتو به جستجو در OlDy پارامترهای ورودی را چک میکنیم و پارامتر اول اجرا میکنیم و آدرس عکس می شود. حال بیاییم بیفتیم با را این پارامترها اجرا کنیم CIU bmp 123456 بعد از اجرا می توان عدد که برا حتی لود می شود و همچنین توسط تابع SetConsoleScreenBufferSize بعد صفحه کنسول به اندازه عکس تغییر می کند.

```
004010DB  .  66 :MOV     WORD PTR SS:[
004010E3  .  8B:MOV     EDX, DWORD PT
004010EA  .  52 :PUSH    EDX
004010EB  .  50 :PUSH    EAX
004010EC  .  FF:CALL     NEAR DWORD PT
004010F2  .  85:TEST     EAX, EAX
```

```
Size = (X=947.,Y=511.)
hConsole = 00000007
SetConsoleScreenBufferSize
```

در ادامه نمی به این پیغام میرسیم :



در ادامه نمی یک حلقه میبینیم خوب ببینیم چکارک انجام میدهد؟

```
004011AD  .  33:XOR     EAX, EAX
004011AF  .  88:MOV     CL, BYTE PTR SS:[ESP+EAX+10]
00401153  .  8B:XOR     CL, DC
00401156  .  88:MOV     BYTE PTR SS:[ESP+EAX+D4], CL
0040115D  .  40:INC     EAX
0040115E  .  83:CMP     EAX, 9
00401161  .  7F:JG      SHORT Green.004011AF
```

```
Address   ASCII dump
0012FF30  G00dJ0b!..?..A@.^@.*@.Z.@.w`@ `.*...@.`..@.`..@.`
```

شبه کد این حلقه بدین صورت است:

```
char Decrypted_String [8];
int Counter = 0;
do
{
  Decrypted_String [Counter] = *(&Encrypted_String + Counter) ^ 0xC;
  ++ Counter;
}
while (Counter < 9 );
```

بعد از پایان حلقه تغییراتی که در حافظه ایجاد شده را در عکس مشاهده میکنیم که ۴کوارکتر می شود G00dJ0b همراه با null در فلوتای رشته. در ادامه نمی یعنی بعد از وارد کردن phrase در کنسول به عکس کمر خواهیم رسید :

```
0040118F  .  8D:LEA     ESI, DWORD PTR SS:[ESP+D4]
00401196  .  8D:LEA     EAX, DWORD PTR SS:[ESP+E8]
```

```
Stack address=0012FF30, (ASCII "G00dJ0b!")
ESI=7C812FD0 (kernel32.GetStdHandle)
```

خوب در ادامه ۲۰۱۱۸ میبینم G00dJ0b با phrase به ۲۰۱۹۲ به phrase که وارد نمودیم. خوب phrase در ادامه تست. حال در ادامه نمی به یک حلقه نسبتا دزهم برمیخوریم که در ابتدا هم چیلنجر کار با پیکسل است.

```
004011EC  .  8D:LEA     ECX, DWORD PTR SS:[ESP+C4]
004011F0  .  51 :PUSH    ECX
004011F1  .  53 :PUSH    EBX
004011F2  .  57 :PUSH    EDI
004011F3  .  FF:CALL     MEM4 DWORD PTR DS:[&GDI32.Ge
```

```
X = 0
Y = 0
hDC = 00010011
GetPixel
```

```
ESI=00000000
Jump from 004197F
```

به فلوتای حلقه می روم میبینم مقدار نگرار حلقه چقدر است :

```
0040196D  .  E8:CALL    Green.004052F4
00401972  .  83:ADD     ESP, 4
00401975  .  8B:MOV     EAX, DWORD PTR SS:[ESP+A8]
0040197C  .  A3:INC     EAX
0040197D  .  3B:CMP     EBX, EAX
0040197F  .  0F:JL      Green.004011F4
```

```
Stack SS:[0012FF04]=00000B2
EAX=00000B2
Jumps from 004013D1, 00401772, 004017A4
```

خوب اینجا EAX و EBX مقایسه می شود و بیستی مقدار EAX به صفر است به EAX برسد. حال مقدار EAX چیست؟ 0x0B2 که در واقع می شود ۲۷۲ که همان ایجاد عکس است:

کارک به این حلقه نداریم چون بیسیار پیچ پیچ است و خود را به نردسر دیمیتاوی:نت

بعد از اتمام حلقه باز به یک حلقه هگ برخورد میکنیم:

```
0040197D  .  3B:CMP     EBX, EAX
0040197F  .  0F:JL      Green.004011F4
00401985  .  68:PUSH    Green.004BF0CC
0040198A  .  E8:CALL    Green.004052F4
0040198F  .  8B:MOV     EAX, DWORD PTR SS:[ESP+B0]
00401996  .  83:ADD     ESP, 4
00401999  .  46:INC     ESI
0040199A  .  3B:CMP     ESI, EAX
0040199C  .  0F:JL      Green.004011E3
```

```
Jump is taken
004011E3=Green.004011E3
```

اطلاعاتی که از این بدست می آید مانند قبلی است ولی این بار مقدار EAX=0x1FE که همان ۵۱۰ است. پس در واقع بو حلقه که طول و عرض عکس را پیدا می کند. بعد از اتمام حلقه دوتایع GetPixel نیاز.

نیاز داریم تا تمامی کارکاف انجام شده و مستندات باقی بعقد پس نمیگذاریم که این دو تابع اجرا شود.

حالا کنسول را بفلوتایین میکنیم تا ببینیم که کارک انجام انجام شده:

بله فلگ رو بدست آوردیم!!! پس کنید سبز رنگ است رهیمی چلیج Green است. پس مشکلی وجود دارد که وقتی کلید را وارد میکنیم ثبت نمی شود!! پس این فلگ نیس!؟ نه ضمن همیشه یه CTF یک تعدد flag مشاهده بشه به کلیه key ها همون کلید!

ادامه نمی را میبریم. در ادامه میبینم که کاراکترهای ASCII printable در حافظه رفت می شود.

```
00401911  > 57:PUSH    EDI
00401912  . FF:CALL    NEAR DWORD PTR DS:[&GDI32.De
00401918  . 8B:MOV     EDX, DWORD PTR SS:[ESP+B0]
00401919  . 52:PUSH    EDX
00401921  . FF:CALL    NEAR DWORD PTR DS:[&GDI32.De
00401927  . B3:MOV     BL, 73
00401929  . B1:MOV     CL, 7A
00401928  . 88:MOV     BYTE PTR SS:[ESP+22], BL
00401928  . 88:MOV     BYTE PTR SS:[ESP+3A], BL
0040192C  . 88:MOV     BYTE PTR SS:[ESP+4E], BL
0040192F  . B8:MOV     AL, 65
00401930  . B2:MOV     DL, 72
00401932  . B3:MOV     BL, 61
00401934  . C6:MOV     BYTE PTR SS:[ESP+16], 68
00401937  . C6:MOV     BYTE PTR SS:[ESP+1B], 69
0040193A  . C6:MOV     BYTE PTR SS:[ESP+1F], 6C
0040193C  . 88:MOV     BYTE PTR SS:[ESP+1F], CL
0040193E  . C6:MOV     BYTE PTR SS:[ESP+29], 3B
00401945  . C6:MOV     BYTE PTR SS:[ESP+2D], 28
0040194A  . C6:MOV     BYTE PTR SS:[ESP+2F], 6D
```

```
hObject = 7C9DE4F0
DeleteObject
```

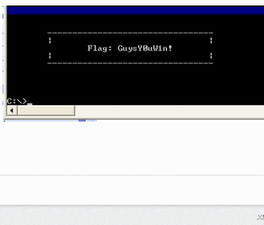
در فلوتا مشاهده میکنیم ببینیم که چیزی در حافظه نوشته شده است که فریگوه است :

```
0040193B  . C6:MOV     BYTE PTR SS:[ESP+98], 67
00401941  . C6:MOV     BYTE PTR SS:[ESP+9B], 2E
00401947  . C6:MOV     BYTE PTR SS:[ESP+9D], 6F
0040194D  . FF:CALL    NEAR DWORD PTR DS:[&KERNEL32.
```

```
SetConsoleScreenBufferInfo
```

```
Address   ASCII dump
0012FF30  hint: you should shift by ten (+2) the Green key characters and
0012FF40  enter it as the second argument to see the correct flag...L.
0012FF50  ..A.B.C.D.E.F.G.H.I.J.K.L.M.N.O.P.Q.R.S.T.U.V.W.X.Y.Z.[.
0012FF60  .\.].^._.`.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.q.r.s.t.u.v.
```

به یک شیت نوشته که کلید کد که در کنسول درجه رابط +2 shift که همان shift right شیت بصورت رفیرنی کاراکترهای فلگ قهبش را ایجاد نمی کند که باید فهمید از روت جدول اسکی رفیستی کاراکترها با شیفت خوب کاراکترها براساس جداول شیفت میخلیم و بعنوان پارامتر دوم به فایل میفرستیم. Green Key ملی براساس با K1hr/1Jdp_PlB8 (که بعد از +2 شیفت بدین صورت خواهد بود K)K1hr/1Jdp_PlB8 *{M=p1L3fr8#~D به معنوان پارامتر دوم قرار میدهیم.

```
C:\>Green.exe c:\1.bmp M"jt13Lfra8#~D:>=
stop please! enter your pass phrase to continue.
G00dJ0b!
```

بعد از اجرا فلگ درست را بدست خواهیم آورد :