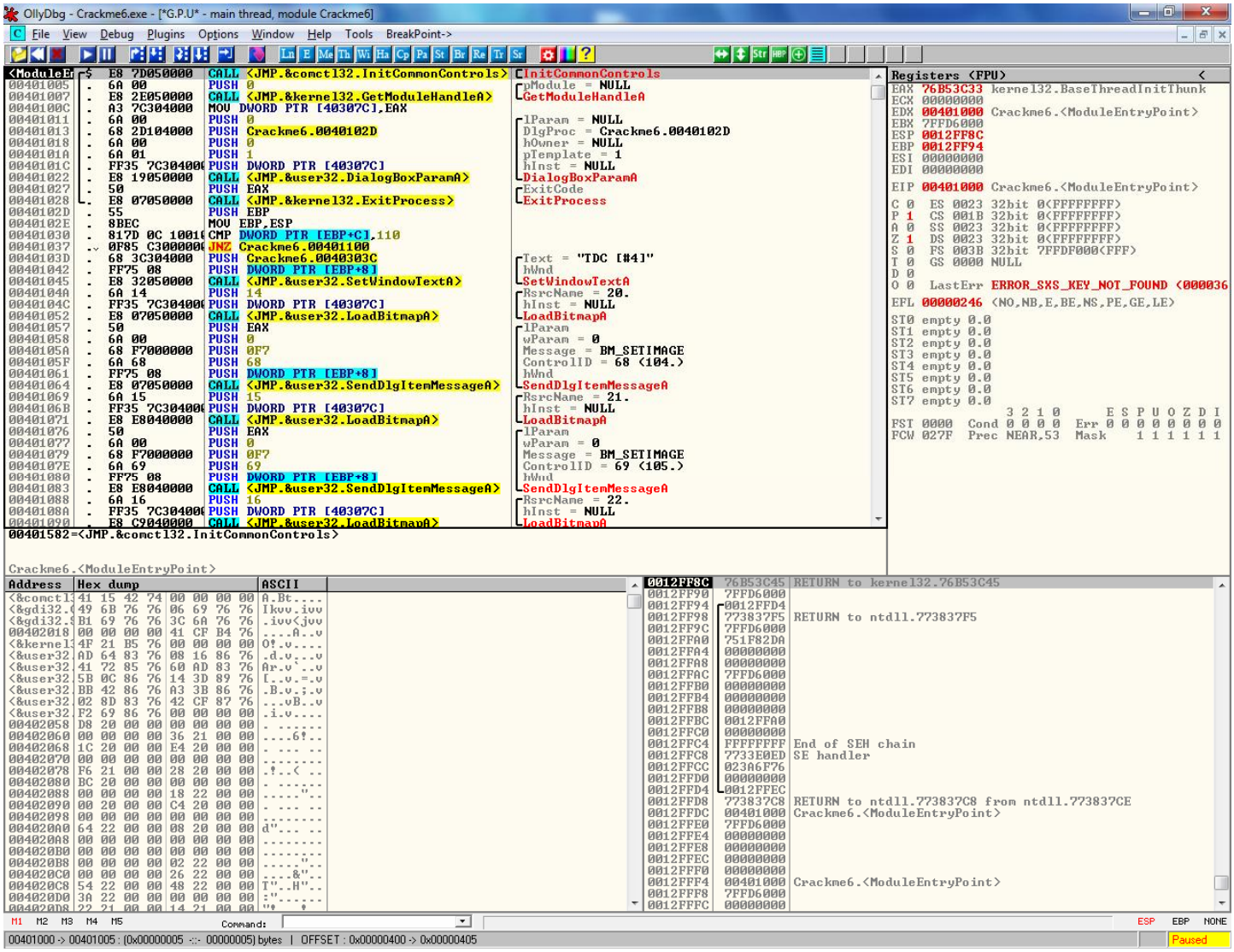
آموزش های R4NDOM -فارسی

قسمت نهم-رشته بی رشته

سلام ،به قسمت نهم از این سری آموزش خوش آمدید،در این قسمت ما با ترفند جدیدی آشنا خواهیم شد ،برنامه ایی را در نظر بگیرید که هیچ گونه رشته ایی به عنوان سرنخ را در خود ندارد!! چکار میکنید؟دراین آموزش روش کرک کردن این نوع را یاد خواهیم گرفت ،برنامه ایی که روی آن عملیات را انجام میدهیم Crackme6 نام دارد و ضمیمه ی این آموزش شده ،بیایید شروع کنیم :

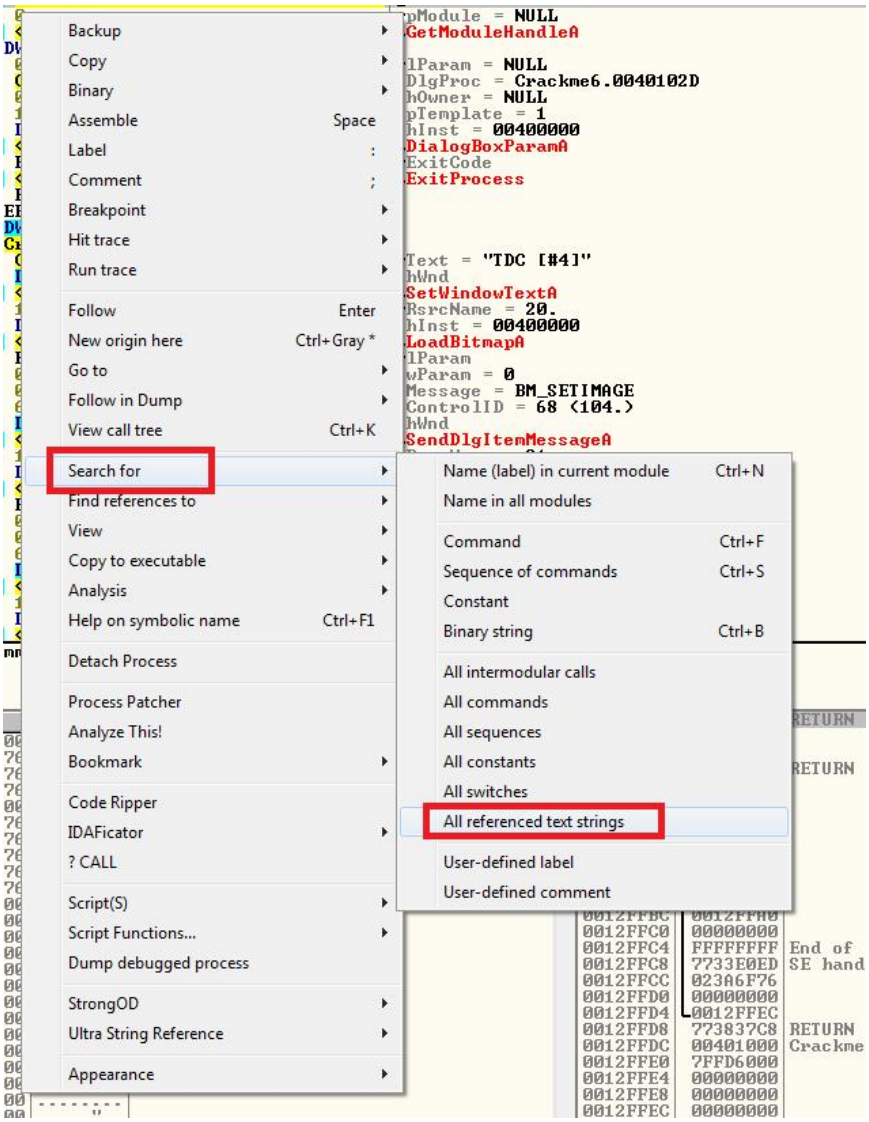
برنامه ی Crackme6 را درون olly بازکنید :

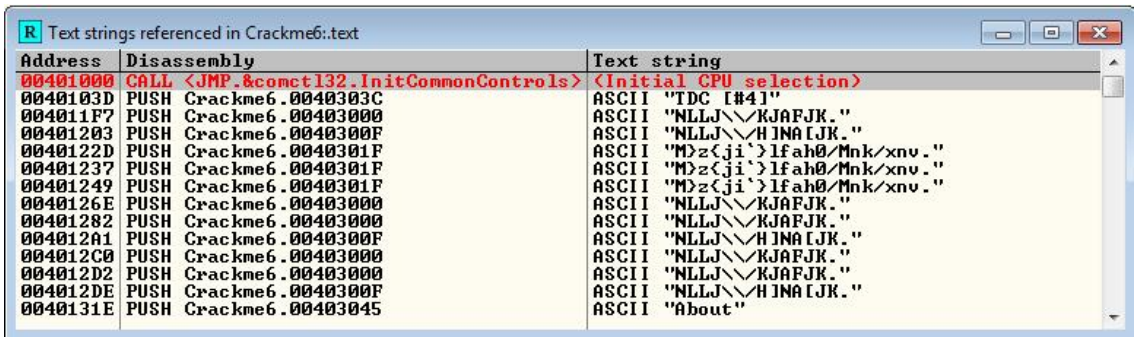


برنامه را اجرا کنید:

یک پسورد دلخواه وارد کنید و کلید Check را بفشارید من عدد 12121212 را وارد کرده ام واین رو نشون داد:

طبق معمول بیایید رشته های مورد استفاده در برنامه را جستجو کنیم "search for Strings" :



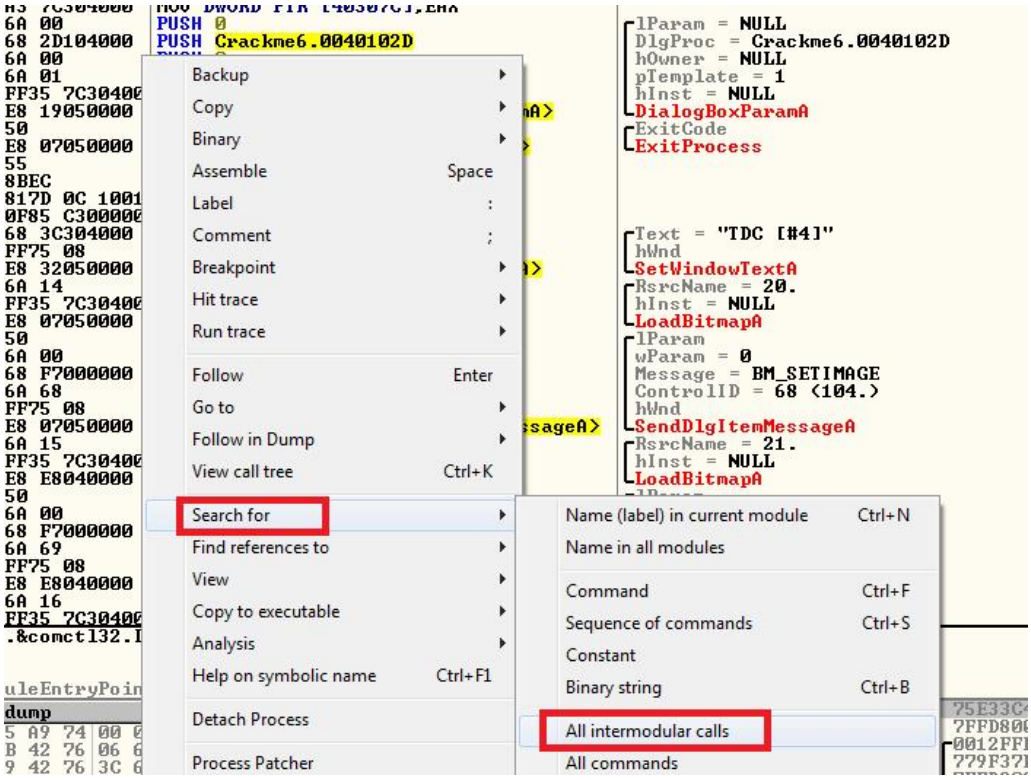


عه!!هیچ چیز بدرد بخوری نیست ،یا شاید هم رمزنگاری / مبهم شده اند ،به یاد داشته باید هیچ وقت ،فقط بر پیدا کردن رشته تکیه نکنید ☺

Intermodular Calls

در اینجا من به شما یک ترفند جدید نشان خواهم داد ،اکثر برنامه های تحت ویندوز از APIs استفاده میکنند ،مثلا از تابع MessageBoxA برای نمایش یک پیام استفاده میشود و یا تابع TerminateProcess موقعی استفاده میشود که برنامه بخواهد خروج کند،از آنجایی که بسیار از برنامه ها از توابع استفاده میکنند ،پس ما هم میتوانیم از آنها به نفع خودمان استفاده کنیم ،مثلا فرض کنید که برنامه ایی رشته ایی تحت عنوان (یوزر و پسورد) را از یک دیالوگ باکس دریافت میکند یا مثلا با استفاده از تایمر بعد از ۱۰ ثانیه پیامی مبنی بر رجیستر نشدن برنامه را نشان میدهد یا مثلا از رجیستری ویندوز برای انجام عملیات خاصی استفاده میکنند ،این ها تماما از توابع ویندوز استفاده میشوند.

OllyDBG یک قابلیت دارد که برای نمایش تمامی API های استفاده شده در برنامه است .برای انجام این کار بروی پنجره ی دی اسمبلی راست کلیک کنید : "Search for" -> "All intermodular calls":



و سپس پنجره ی intermodular calls را نمایش میدهد :

Found intermodular calls

Address	Disassembly	Destination
00401000	CALL <JMP.&comctl32.InitCommonControls>	<Initial CPU selection>
00401007	CALL <JMP.&kernel32.GetModuleHandleA>	kernel32.GetModuleHandleA
00401022	CALL <JMP.&user32.DialogBoxParamA>	user32.DialogBoxParamA
00401028	CALL <JMP.&kernel32.ExitProcess>	kernel32.ExitProcess
00401045	CALL <JMP.&user32.SetWindowTextA>	user32.SetWindowTextA
00401052	CALL <JMP.&user32.LoadBitmapA>	user32.LoadBitmapA
00401064	CALL <JMP.&user32.SendDlgItemMessageA>	user32.SendDlgItemMessageA
00401071	CALL <JMP.&user32.LoadBitmapA>	user32.LoadBitmapA
00401083	CALL <JMP.&user32.SendDlgItemMessageA>	user32.SendDlgItemMessageA
00401090	CALL <JMP.&user32.LoadBitmapA>	user32.LoadBitmapA
004010A2	CALL <JMP.&user32.SendDlgItemMessageA>	user32.SendDlgItemMessageA
004010AF	CALL <JMP.&user32.LoadIconA>	user32.LoadIconA
004010BF	CALL <JMP.&user32.SendMessageA>	user32.SendMessageA
004010C9	CALL <JMP.&user32.GetDlgItem>	user32.GetDlgItem
004010EA	CALL <JMP.&user32.SetWindowTextA>	user32.SetWindowTextA
0040110E	CALL <JMP.&gdi32.CreateSolidBrush>	gdi32.CreateSolidBrush
0040112D	CALL <JMP.&gdi32.SetTextColor>	gdi32.SetTextColor
00401137	CALL <JMP.&gdi32.SetBkMode>	gdi32.SetBkMode
00401141	CALL <JMP.&gdi32.CreateSolidBrush>	gdi32.CreateSolidBrush
0040114A	CALL <JMP.&gdi32.SetBkColor>	gdi32.SetBkColor
00401154	CALL <JMP.&gdi32.CreateSolidBrush>	gdi32.CreateSolidBrush
00401173	CALL <JMP.&gdi32.SetTextColor>	gdi32.SetTextColor
00401180	CALL <JMP.&gdi32.SetBkColor>	gdi32.SetBkColor
0040118A	CALL <JMP.&gdi32.CreateSolidBrush>	gdi32.CreateSolidBrush
004011A1	CALL <JMP.&user32.ReleaseCapture>	user32.ReleaseCapture
004011B2	CALL <JMP.&user32.SendMessageA>	user32.SendMessageA
004011E1	CALL <JMP.&user32.DialogBoxParamA>	user32.DialogBoxParamA
00401242	CALL <JMP.&user32.SetWindowTextA>	user32.SetWindowTextA
00401264	CALL <JMP.&user32.GetDlgItemTextA>	user32.GetDlgItemTextA
00401279	CALL <JMP.&user32.SetWindowTextA>	user32.SetWindowTextA
0040128D	CALL <JMP.&user32.SetWindowTextA>	user32.SetWindowTextA
004012AC	CALL <JMP.&user32.SetWindowTextA>	user32.SetWindowTextA
004012B9	CALL <JMP.&user32.EnableWindow>	user32.EnableWindow
004012CB	CALL <JMP.&user32.SetWindowTextA>	user32.SetWindowTextA
004012F5	CALL <JMP.&user32.EndDialog>	user32.EndDialog
00401307	CALL <JMP.&user32.EndDialog>	user32.EndDialog
00401326	CALL <JMP.&user32.SetWindowTextA>	user32.SetWindowTextA
00401333	CALL <JMP.&user32.LoadBitmapA>	user32.LoadBitmapA
00401345	CALL <JMP.&user32.SendDlgItemMessageA>	user32.SendDlgItemMessageA
0040135D	CALL <JMP.&gdi32.CreateSolidBrush>	gdi32.CreateSolidBrush
0040137C	CALL <JMP.&gdi32.SetTextColor>	gdi32.SetTextColor
00401386	CALL <JMP.&gdi32.SetBkMode>	gdi32.SetBkMode
00401390	CALL <JMP.&gdi32.CreateSolidBrush>	gdi32.CreateSolidBrush
00401399	CALL <JMP.&gdi32.SetBkColor>	gdi32.SetBkColor
004013A3	CALL <JMP.&gdi32.CreateSolidBrush>	gdi32.CreateSolidBrush
004013BF	CALL <JMP.&gdi32.SetTextColor>	gdi32.SetTextColor
004013CC	CALL <JMP.&gdi32.SetBkColor>	gdi32.SetBkColor
004013D6	CALL <JMP.&gdi32.CreateSolidBrush>	gdi32.CreateSolidBrush
004013EA	CALL <JMP.&user32.ReleaseCapture>	user32.ReleaseCapture
004013FB	CALL <JMP.&user32.SendMessageA>	user32.SendMessageA
00401416	CALL <JMP.&user32.EndDialog>	user32.EndDialog

اولین چیزی که مهم است مرتب سازی بر اساس مقصد/حروف الفبا است :

- [Found intermodular calls]

File View Debug Plugins Options Window Help Tools BreakPoint->

Address	Disassembly	Destination
0040110E	CALL <JMP.&gdi32.CreateSolidBrush>	gdi32.CreateSolidBrush
00401141	CALL <JMP.&gdi32.CreateSolidBrush>	gdi32.CreateSolidBrush
00401154	CALL <JMP.&gdi32.CreateSolidBrush>	gdi32.CreateSolidBrush
0040118A	CALL <JMP.&gdi32.CreateSolidBrush>	gdi32.CreateSolidBrush
0040135D	CALL <JMP.&gdi32.CreateSolidBrush>	gdi32.CreateSolidBrush
00401390	CALL <JMP.&gdi32.CreateSolidBrush>	gdi32.CreateSolidBrush
004013A3	CALL <JMP.&gdi32.CreateSolidBrush>	gdi32.CreateSolidBrush
004013D6	CALL <JMP.&gdi32.CreateSolidBrush>	gdi32.CreateSolidBrush
00401022	CALL <JMP.&user32.DialogBoxParamA>	user32.DialogBoxParamA
004011E1	CALL <JMP.&user32.DialogBoxParamA>	user32.DialogBoxParamA
004012B9	CALL <JMP.&user32.EnableWindow>	user32.EnableWindow
004012F5	CALL <JMP.&user32.EndDialog>	user32.EndDialog
00401307	CALL <JMP.&user32.EndDialog>	user32.EndDialog
00401416	CALL <JMP.&user32.EndDialog>	user32.EndDialog
00401028	CALL <JMP.&kernel32.ExitProcess>	kernel32.ExitProcess
004010C9	CALL <JMP.&user32.GetDlgItem>	user32.GetDlgItem
00401264	CALL <JMP.&user32.GetDlgItemTextA>	user32.GetDlgItemTextA
00401007	CALL <JMP.&kernel32.GetModuleHandleA>	kernel32.GetModuleHandleA
00401000	CALL <JMP.&comctl32.InitCommonControls>	<Initial CPU selection>
00401052	CALL <JMP.&user32.LoadBitmapA>	user32.LoadBitmapA
00401071	CALL <JMP.&user32.LoadBitmapA>	user32.LoadBitmapA
00401090	CALL <JMP.&user32.LoadBitmapA>	user32.LoadBitmapA
00401333	CALL <JMP.&user32.LoadBitmapA>	user32.LoadBitmapA
004010AF	CALL <JMP.&user32.LoadIconA>	user32.LoadIconA
004011A1	CALL <JMP.&user32.ReleaseCapture>	user32.ReleaseCapture
004013EA	CALL <JMP.&user32.ReleaseCapture>	user32.ReleaseCapture
00401064	CALL <JMP.&user32.SendDlgItemMessageA>	user32.SendDlgItemMessageA
00401083	CALL <JMP.&user32.SendDlgItemMessageA>	user32.SendDlgItemMessageA
004010A2	CALL <JMP.&user32.SendDlgItemMessageA>	user32.SendDlgItemMessageA
00401345	CALL <JMP.&user32.SendDlgItemMessageA>	user32.SendDlgItemMessageA
004010BF	CALL <JMP.&user32.SendMessageA>	user32.SendMessageA
004011B2	CALL <JMP.&user32.SendMessageA>	user32.SendMessageA
004013FB	CALL <JMP.&user32.SendMessageA>	user32.SendMessageA
0040114A	CALL <JMP.&gdi32.SetBkColor>	gdi32.SetBkColor
00401180	CALL <JMP.&gdi32.SetBkColor>	gdi32.SetBkColor
00401399	CALL <JMP.&gdi32.SetBkColor>	gdi32.SetBkColor
004013CC	CALL <JMP.&gdi32.SetBkColor>	gdi32.SetBkColor
00401137	CALL <JMP.&gdi32.SetBkMode>	gdi32.SetBkMode
00401386	CALL <JMP.&gdi32.SetBkMode>	gdi32.SetBkMode
0040112D	CALL <JMP.&gdi32.SetTextColor>	gdi32.SetTextColor
00401173	CALL <JMP.&gdi32.SetTextColor>	gdi32.SetTextColor
0040137C	CALL <JMP.&gdi32.SetTextColor>	gdi32.SetTextColor
004013BF	CALL <JMP.&gdi32.SetTextColor>	gdi32.SetTextColor
00401045	CALL <JMP.&user32.SetWindowTextA>	user32.SetWindowTextA
004010EA	CALL <JMP.&user32.SetWindowTextA>	user32.SetWindowTextA
00401242	CALL <JMP.&user32.SetWindowTextA>	user32.SetWindowTextA
00401279	CALL <JMP.&user32.SetWindowTextA>	user32.SetWindowTextA
0040128D	CALL <JMP.&user32.SetWindowTextA>	user32.SetWindowTextA
004012AC	CALL <JMP.&user32.SetWindowTextA>	user32.SetWindowTextA
004012CB	CALL <JMP.&user32.SetWindowTextA>	user32.SetWindowTextA
00401326	CALL <JMP.&user32.SetWindowTextA>	user32.SetWindowTextA

برای مرتب سازی بر حسب "مقصد" کلیک کنید

همانطور که میبینید تمامی API هایی که توسط کرک می استفاده میشوند را نمایش داده ، البته این برنامه کوچکی است و تعداد این API ها کم است در برنامه های بزرگ تعداد بسیار زیاد است ، به هر حال این لیست بسیار بسیار کمک کننده است و در حقیقت

شیرازه ی برنامه را مشخص میکند ،هر حرکتی اعم از لود کردن عکس ،نوشتن در فایل ،دریافت اطلاعات ،تغییر رنگ و... را میتوان فهمید

در برنامه های بزرگتر این اطلاعت ارزشمندتر و زیاد تر میشوند ،آنها به اطلاعاتی نظیر :بازیابی و ذخیر اطلاعات از رجیستری ، آیا رجیستر کردن اطلاعات باید از وب سایت باشد ، آیا برای ریجیستر کردن نیاز به خواندن و نوشتن درون یک فایل مشخص نیاز است و....و اوج اهمیت موقعیی است که برنامه "پک یا مبهم" شده باشد.

تعدادی از مهمترین و کاربردی ترین API های مورد استفاده در طرح های حفاظتی :

DialogBoxParamA
GetDlgItem
GetDlgItemInt
GetDlgTextA
GetWindowTextA
GetWindowWord
LoadStringA
lstrcmpA
wsprintfA
MessageBeep
MessageBoxA
MessageBoxExA
SendMessageA

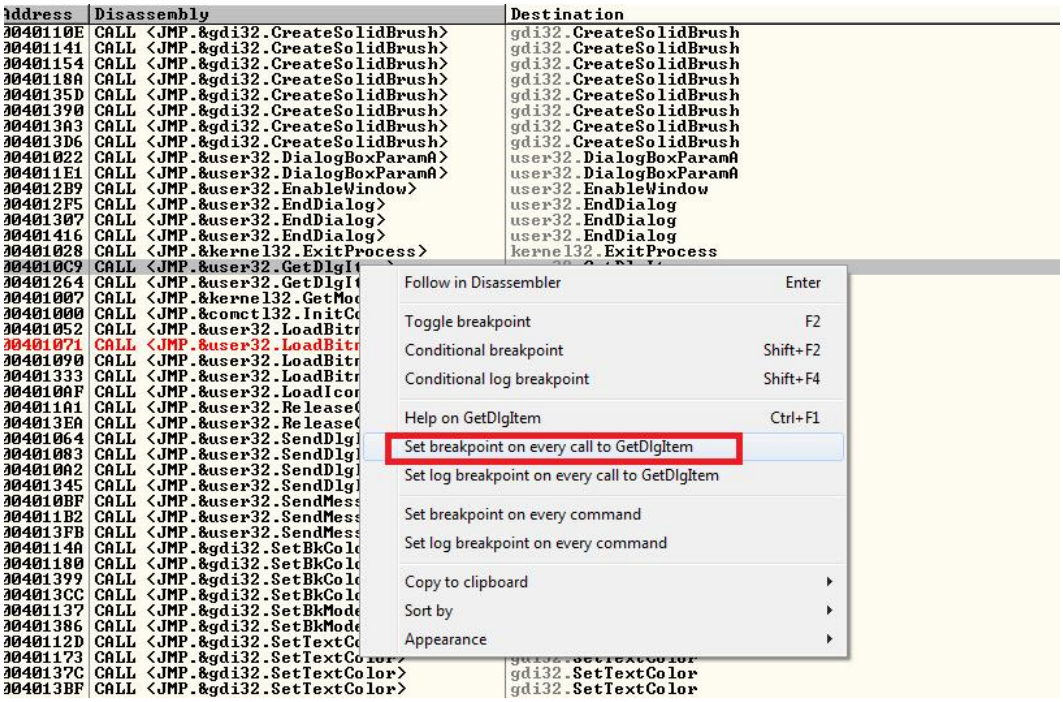
SendDlgItemMessageA
ReadFile
WriteFile
CreateFileA
GetPrivateProfileIntA
WritePrivateProfileStringA
GetPrivateProfileStringA

البته این API ها همه ی طرح های حفاظتی را پوشش نمیدهند ولی بسیار از آنها را پوشش میدهد،اما در این مثال این کرک می از این توابع استفاده میکند:

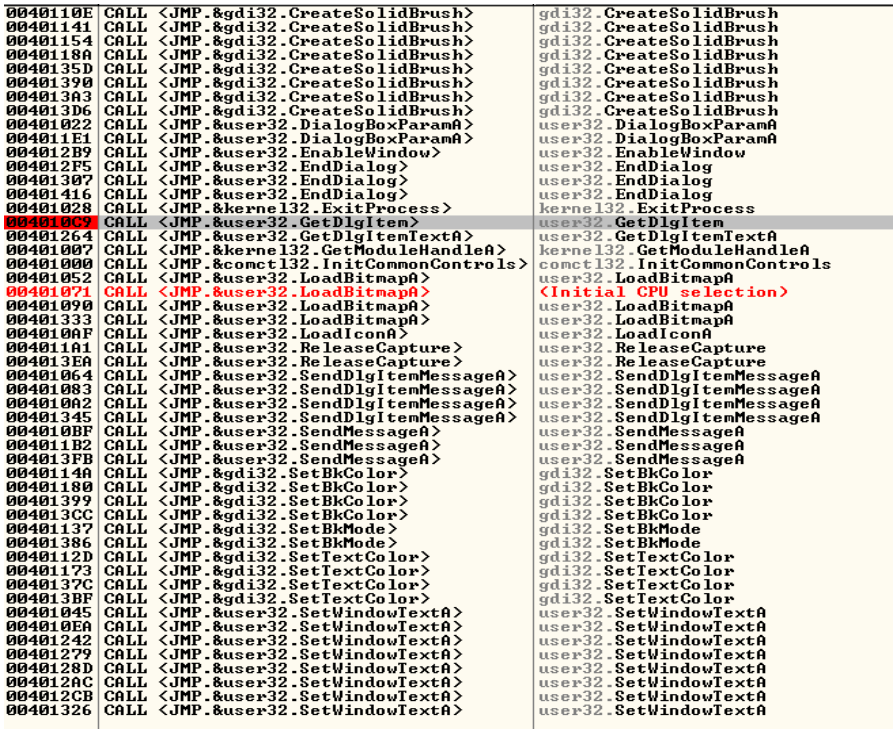
GetDlgItemTextA
GetWindowTextA
lstrcmpA
GetPrivateProfileStringA
GetPrivateProfileIntA
RegQueryValueExA
WritePrivateProfileStringA
GetPrivateProfileIntA

اگر به روی همین ۸ تابع تمرکز کنیم به راحتی میتوانیم اکثر موارد مربوط به امنیت برنامه را متوجه شویم ،در ضمن یادتان نرود میتوانید از پلاگین "help on symbolic name" هم کمک بگیرید.

وقتی به لیست API ها نگاه میکنید متوجه دوتابع میشویم که Crackme از آن استفاده میکند: GetDlgItem و GetDlgItemTextA ،آنچه در باره ی این دو تابع باید بدانیم این است که این دو تابع مسئول دریافت هرگونه متنی است که درون دیالوگ باکس وارد شده باشد،بسیار خوب،این جمله در مورد کرک می ما یعنی چه؟ ☺ بله درست است ،دریافت پسورد ☺ خب ما میخواهیم به Ollydbg اعلام کنیم که هر وقت برنامه از یکی از این دوتابع استفاده کرد به ما اطلاع دهد ،این کار را با گذاشتن BP بروی تابع GetDlgItem انجام میدهیم :



و میبینید که Ollyبروی تابع مورد نظر یک BP گذاشته :



همین کار را برای تابع GetDlgItemTextA را هم انجام میدهیم :

```
0040110E  CALL <JMP.&gdi32.CreateSolidBrush>        gdi32.CreateSolidBrush
00401141  CALL <JMP.&gdi32.CreateSolidBrush>        gdi32.CreateSolidBrush
00401154  CALL <JMP.&gdi32.CreateSolidBrush>        gdi32.CreateSolidBrush
0040118A  CALL <JMP.&gdi32.CreateSolidBrush>        gdi32.CreateSolidBrush
0040135D  CALL <JMP.&gdi32.CreateSolidBrush>        gdi32.CreateSolidBrush
00401390  CALL <JMP.&gdi32.CreateSolidBrush>        gdi32.CreateSolidBrush
004013A3  CALL <JMP.&gdi32.CreateSolidBrush>        gdi32.CreateSolidBrush
004013D6  CALL <JMP.&gdi32.CreateSolidBrush>        gdi32.CreateSolidBrush
00401022  CALL <JMP.&user32.DialogBoxParamA>        user32.DialogBoxParamA
004011E1  CALL <JMP.&user32.DialogBoxParamA>        user32.DialogBoxParamA
004012B9  CALL <JMP.&user32.EnableWindow>           user32.EnableWindow
004012F5  CALL <JMP.&user32.EndDialog>              user32.EndDialog
00401307  CALL <JMP.&user32.EndDialog>              user32.EndDialog
00401416  CALL <JMP.&user32.EndDialog>              user32.EndDialog
00401028  CALL <JMP.&kernel32.ExitProcess>          kernel32.ExitProcess
004010C9  CALL <JMP.&user32.GetDlgItem>             user32.GetDlgItem
00401264  CALL <JMP.&user32.GetDlgItemTextA>        user32.GetDlgItemTextA
00401007  CALL <JMP.&kernel32.GetModuleHandleA>     kernel32.GetModuleHandleA
00401000  CALL <JMP.&comctl32.InitCommonControls>   comctl32.InitCommonControls
00401052  CALL <JMP.&user32.LoadBitmapA>            user32.LoadBitmapA
00401071  CALL <JMP.&user32.LoadBitmapA>            (Initial CPU selection)
00401090  CALL <JMP.&user32.LoadBitmapA>            user32.LoadBitmapA
00401333  CALL <JMP.&user32.LoadBitmapA>            user32.LoadBitmapA
004010AF  CALL <JMP.&user32.LoadIconA>              user32.LoadIconA
004011A1  CALL <JMP.&user32.ReleaseCapture>         user32.ReleaseCapture
004013EA  CALL <JMP.&user32.ReleaseCapture>         user32.ReleaseCapture
00401064  CALL <JMP.&user32.SendDlgItemMessageA>    user32.SendDlgItemMessageA
00401083  CALL <JMP.&user32.SendDlgItemMessageA>    user32.SendDlgItemMessageA
004010A2  CALL <JMP.&user32.SendDlgItemMessageA>    user32.SendDlgItemMessageA
00401345  CALL <JMP.&user32.SendDlgItemMessageA>    user32.SendDlgItemMessageA
004010BF  CALL <JMP.&user32.SendMessageA>           user32.SendMessageA
004011B2  CALL <JMP.&user32.SendMessageA>           user32.SendMessageA
004013FB  CALL <JMP.&user32.SendMessageA>           user32.SendMessageA
0040114A  CALL <JMP.&gdi32.SetBkColor>              gdi32.SetBkColor
00401180  CALL <JMP.&gdi32.SetBkColor>              gdi32.SetBkColor
00401399  CALL <JMP.&gdi32.SetBkColor>              gdi32.SetBkColor
004013CC  CALL <JMP.&gdi32.SetBkColor>              gdi32.SetBkColor
00401137  CALL <JMP.&gdi32.SetBkMode>               gdi32.SetBkMode
00401386  CALL <JMP.&gdi32.SetBkMode>               gdi32.SetBkMode
0040112D  CALL <JMP.&gdi32.SetTextColor>            gdi32.SetTextColor
00401173  CALL <JMP.&gdi32.SetTextColor>            gdi32.SetTextColor
0040137C  CALL <JMP.&gdi32.SetTextColor>            gdi32.SetTextColor
004013BF  CALL <JMP.&gdi32.SetTextColor>            gdi32.SetTextColor
00401045  CALL <JMP.&user32.SetWindowTextA>         user32.SetWindowTextA
004010EA  CALL <JMP.&user32.SetWindowTextA>         user32.SetWindowTextA
00401242  CALL <JMP.&user32.SetWindowTextA>         user32.SetWindowTextA
00401279  CALL <JMP.&user32.SetWindowTextA>         user32.SetWindowTextA
0040128D  CALL <JMP.&user32.SetWindowTextA>         user32.SetWindowTextA
004012AC  CALL <JMP.&user32.SetWindowTextA>         user32.SetWindowTextA
004012CB  CALL <JMP.&user32.SetWindowTextA>         user32.SetWindowTextA
00401326  CALL <JMP.&user32.SetWindowTextA>         user32.SetWindowTextA
```

حال هر موقع که از این توابع استفاده شود ،Ollyبه محض برخورد با آن متوقف شده ،بیایید برنامه را ری استارت کنید و دکمـه F9 را ، Olly بروی تابع GetDlgItemمتوقف شده :

```
0040109F  .  FF75 08        PUSH DWORD PTR [EBP+8]                 hWnd
004010A2  .  E8 C9040000    CALL <JMP.&user32.SendDlgItemMessageA> SendDlgItemMessageA
004010A7  .  6A 64          PUSH 64                                RsrcName = 100.
004010A9  .  FF35 7C304000  PUSH DWORD PTR [40307C]                hInst = 00400000
004010AF  .  E8 B0040000    CALL <JMP.&user32.LoadIconA>           LoadIconA
004010B4  .  50             PUSH EAX                               lParam
004010B5  .  6A 01          PUSH 1                                 wParam = 1
004010B7  .  68 80000000    PUSH 80                                Message = WM_SETICON
004010BC  .  FF75 08        PUSH DWORD PTR [EBP+8]                 hWnd
004010BF  .  E8 B2040000    CALL <JMP.&user32.SendMessageA>        SendMessageA
004010C4  .  6A 6B          PUSH 6B                                ControlID = 6B (107.)
004010C6  .  FF75 08        PUSH DWORD PTR [EBP+8]                 hWnd
004010C9  .  E8 84040000    CALL <JMP.&user32.GetDlgItem>          GetDlgItem
004010CE  .  A3 80304000    MOV DWORD PTR [403080],EAX
004010D3  .  6A 12          PUSH 12
004010D5  .  68 69304000    PUSH Crackme6.00403069
004010DA  .  E8 3C040000    CALL Crackme6.0040151B
004010DF  .  68 69304000    PUSH Crackme6.00403069                 Text = "U`z}/fa.z{/.cjn!j!"
004010E4  .  FF35 80304000  PUSH DWORD PTR [403080]                hWnd = NULL
004010EA  .  E8 8D040000    CALL <JMP.&user32.SetWindowTextA>      SetWindowTextA
004010EF  .  6A 12          PUSH 12
```

از آنجایی که نه هنوز برنامه بالا آمده و نه هنوز چیزی وارد کرده ایم از این قسمت عبور میکنیم بافشردن کلید F9 :

```
0040109F  .   FF75 08        PUSH DWORD PTR [EBP+8]                  hWnd
004010A2  .   E8 C9040000    CALL <JMP.&user32.SendDlgItemMessageA>  SendDlgItemMessageA
004010A7  .   6A 64          PUSH 64                                 RsrcName = 100.
004010A9  .   FF35 7C304000  PUSH DWORD PTR [40307C]                 hInst = 00400000
004010AF  .   E8 B0040000    CALL <JMP.&user32.LoadIconA>            LoadIconA
004010B4  .   50             PUSH EAX                                lParam
004010B5  .   6A 01          PUSH 1                                  wParam = 1
004010B7  .   68 80000000    PUSH 80                                 Message = WM_SETICON
004010BC  .   FF75 08        PUSH DWORD PTR [EBP+8]                  hWnd
004010BF  .   E8 B2040000    CALL <JMP.&user32.SendMessageA>         SendMessageA
004010C4  .   6A 6B          PUSH 6B                                 ControlID = 6B (107.)
004010C6  .   FF75 08        PUSH DWORD PTR [EBP+8]                  hWnd
004010C9  .   E8 84040000    CALL <JMP.&user32.GetDlgItem>           GetDlgItem
004010CE  .   A3 80304000    MOV DWORD PTR [403080],EAX
004010D3  .   6A 12          PUSH 12
004010D5  .   68 69304000    PUSH Crackme6.00403069
004010DA  .   E8 3C040000    CALL Crackme6.0040151B
004010DF  .   68 69304000    PUSH Crackme6.00403069                  Text = "V`z>/fa.z</.cjn!j!"
004010E4  .   FF35 80304000  PUSH DWORD PTR [403080]                 hWnd = NULL
004010EA  .   E8 8D040000    CALL <JMP.&user32.SetWindowTextA>       SetWindowTextA
004010EF  .   6A 12          PUSH 12
004010F1  .   68 69304000    PUSH Crackme6.00403069
004010F6  .   E8 20040000    CALL Crackme6.0040151B
004010FB  .v  E9 0C020000    JMP Crackme6.0040130C
00401100  >   817D 0C 3601   CMP DWORD PTR [EBP+C]
00401107  .v  75 13          JNZ SHORT Crackme6.00
00401109  .   68 00DD0000    PUSH 0DD00                              ..221..0.>
0040110E  .   E8 75040000    CALL <JMP.&gdi32.Crea                   ush
00401113  .   C9             LEAVE
00401114  .   C2 1000        RET 10
00401117  .v  E9 F0010000    JMP Crackme6.0040130C
0040111C  >   817D 0C 3801   CMP DWORD PTR [EBP+C],
00401123  .v  75 3D          JNZ SHORT Crackme6.00401162
00401125  .   68 FFFFFF00    PUSH 0FFFFFF                            Color = <WHITE>
0040112A  .   FF75 10        PUSH DWORD PTR [EBP+10]                 hDC
0040112D  .   E8 68040000    CALL <JMP.&gdi32.SetTextColor>          SetTextColor
00401132  .   6A 01          PUSH 1                                  BkMode = TRANSPARENT
00401134  .   FF75 10        PUSH DWORD PTR [EBP+10]                 hDC
```


حالا یک پسورد دلخواه وارد کنید من "1212121212" را وارد میکنم وکلید Check را فشار دهید، olly متوقف شد ،به این معنی است که تابع GetDlgItemTextA فراخوانی شده است :

```
00401253  .v  E9 B4000000    JMP Crackme6.0040130C
00401258  >   6A 0C          PUSH 0C                                 Count = C (12.)
0040125A  .   68 5D304000    PUSH Crackme6.0040305D                  Buffer = Crackme6.0040305D
0040125F  .   6A 6B          PUSH 6B                                 ControlID = 6B (107.)
00401261  .   FF75 08        PUSH DWORD PTR [EBP+8]                  hWnd
00401264  .   E8 EF020000    CALL <JMP.&user32.GetDlgItemTextA>      GetDlgItemTextA
00401269  .   83F8 0B        CMP EAX,0B
0040126C  .v  72 10          JB SHORT Crackme6.0040127E
0040126E  .   68 00304000    PUSH Crackme6.00403000                  Text = "ACCESS DENIED!"
00401273  .   FF35 80304000  PUSH DWORD PTR [403080]                 hWnd = 0058051A (class='Edit',parent=00C60438
00401279  .   E8 FE020000    CALL <JMP.&user32.SetWindowTextA>       SetWindowTextA
0040127E  >   85C0           TEST EAX,EAX
00401280  .v  75 10          JNZ SHORT Crackme6.00401292
00401282  .   68 00304000    PUSH Crackme6.00403000                  Text = "ACCESS DENIED!"
00401287  .   FF35 80304000  PUSH DWORD PTR [403080]                 hWnd = 0058051A (class='Edit',parent=00C60438
0040128D  .   E8 EA020000    CALL <JMP.&user32.SetWindowTextA>       SetWindowTextA
00401292  >   50             PUSH EAX
00401293  .   68 5D304000    PUSH Crackme6.0040305D
00401298  .   E8 84010000    CALL Crackme6.00401421
0040129D  .   0BC0           OR EAX,EAX
0040129F  .v  75 1F          JNZ SHORT Crackme6.004012C0
004012A1  .   68 0F304000    PUSH Crackme6.0040300F                  Text = "ACCESS GRANTED!"
004012A6  .   FF35 80304000  PUSH DWORD PTR [403080]                 hWnd = 0058051A (class='Edit',parent=00C60438
004012AC  .   E8 CB020000    CALL <JMP.&user32.SetWindowTextA>       SetWindowTextA
004012B1  .   6A 00          PUSH 0                                  Enable = FALSE
004012B3  .   FF35 80304000  PUSH DWORD PTR [403080]                 hWnd = 0058051A (class='Edit',parent=00C60438
004012B9  .   E8 88020000    CALL <JMP.&user32.EnableWindow>         EnableWindow
004012BE  .v  EB 10          JMP SHORT Crackme6.004012D0
004012C0  >   68 00304000    PUSH Crackme6.00403000                  Text = "ACCESS DENIED!"
004012C5  .   FF35 80304000  PUSH DWORD PTR [403080]                 hWnd = 0058051A (class='Edit',parent=00C60438
004012CB  .   E8 AC020000    CALL <JMP.&user32.SetWindowTextA>       SetWindowTextA
004012D0  >   6A 0E          PUSH 0E
004012D2  .   68 00304000    PUSH Crackme6.00403000                  ASCII "ACCESS DENIED!"
004012D7  .   E8 3F020000    CALL Crackme6.0040151B
004012DC  .   6A 0F          PUSH 0F
004012DE  .   68 0F304000    PUSH Crackme6.0040300F                  ASCII "ACCESS GRANTED!"
004012E3  .   E8 33020000    CALL Crackme6.0040151B
004012E8  .   EB 22          JMP SHORT Crackme6.0040130C
```

همانطور که میبینید دقیقا جایی مناسبی هستیم و رشته هایی را میبینیم که Olly در جستجوی آنها موفق نبود

کرک برنامه

خب بیایید سریع باشیم ،شما اکنون چیز های زیادی میدانید ،اگر به دو خط پایین تر نگاه کنیم دستور (JB)را میبینیم که میتواند مـا را از نمایش رشته "ACCESS DENIED" عبور دهد

```
00401261  .   FF75 08        PUSH DWORD PTR SS:[EBP+8]          hWnd = 00090DC8 ('TDC [#4]',class='#32770')
00401264  .   E8 EF020000    CALL <JMP.&user32.GetDlgItemTextA>  GetDlgItemTextA
00401269  .   83F8 0B        CMP EAX,0B
0040126C  .v  72 10          JB SHORT Crackme6.0040127E
0040126E  .   68 00304000    PUSH Crackme6.00403000              Text = "ACCESS DENIED!"
00401273  .   FF35 80304000  PUSH DWORD PTR DS:[403080]          hWnd = 00120DC2 (class='Edit',parent=00090DC8
00401279  .   E8 FE020000    CALL <JMP.&user32.SetWindowTextA>   SetWindowTextA
0040127E  >   85C0           TEST EAX,EAX                        Crackme6.0040300F
00401280  .v  75 10          JNZ SHORT Crackme6.00401292
00401282  .   68 00304000    PUSH Crackme6.00403000              Text = "ACCESS DENIED!"
```

پس از این دستور، یک دستور (JNZ) وجود دارد که این هم مانند قبلی میتواند مـا را از نمـایش "ACCESS DENIED" دوم نجـات دهد،پس باید از هردو دستور همواره پرش صورت گیرد :

```
0040125A  .   68 5D304000    PUSH Crackme6.0040305D              Buffer = Crackme6.0040305D
0040125F  .   6A 6B          PUSH 6B                             ControlID = 6B (107.)
00401261  .   FF75 08        PUSH DWORD PTR SS:[EBP+8]           hWnd = 00090DC8 ('TDC [#4]',class='#32770')
00401264  .   E8 EF020000    CALL <JMP.&user32.GetDlgItemTextA>  GetDlgItemTextA
00401269  .   83F8 0B        CMP EAX,0B
0040126C  .v  72 10          JB SHORT Crackme6.0040127E
0040126E  .   68 00304000    PUSH Crackme6.00403000              Text = "ACCESS DENIED!"
00401273  .   FF35 80304000  PUSH DWORD PTR DS:[403080]          hWnd = 00120DC2 (class='Edit',parent=00090DC8)
00401279  .   E8 FE020000    CALL <JMP.&user32.SetWindowTextA>   SetWindowTextA
0040127E  >   85C0           TEST EAX,EAX                        Crackme6.0040300F
00401280  .v  75 10          JNZ SHORT Crackme6.00401292
00401282  .   68 00304000    PUSH Crackme6.00403000              Text = "ACCESS DENIED!"
00401287  .   FF35 80304000  PUSH DWORD PTR DS:[403080]          hWnd = 00120DC2 (class='Edit',parent=00090DC8)
0040128D  .   E8 EA020000    CALL <JMP.&user32.SetWindowTextA>   SetWindowTextA
00401292  >   50             PUSH EAX                            Crackme6.0040300F
00401293  .   68 5D304000    PUSH Crackme6.0040305D
00401298  .   E8 84010000    CALL Crackme6.00401421
0040129D  .   0BC0           OR EAX,EAX                          Crackme6.0040300F
0040129F  .v  75 1F          JNZ SHORT Crackme6.004012C0
004012A1  .   68 0F304000    PUSH Crackme6.0040300F              Text = "ACCESS GRANTED!"
004012A6  .   FF35 80304000  PUSH DWORD PTR DS:[403080]          hWnd = 00120DC2 (class='Edit',parent=00090DC8)
```

بیایید انجامش دهیم کلید F8 را بفشارید و سپس وقتی بروی دستور JB قرار گرفتید مقدار پرچم (C) را به 1 تغییر داده :

```
0040125F  .   6A 6B          PUSH 6B                             ControlID = 6B (107.)
00401261  .   FF75 08        PUSH DWORD PTR [EBP+8]              hWnd
00401264  .   E8 EF020000    CALL <JMP.&user32.GetDlgItemTextA>  GetDlgItemTextA
00401269  .   83F8 0B        CMP EAX,0B
0040126C  .v  72 10          JB SHORT Crackme6.0040127E
0040126E  .   68 00304000    PUSH Crackme6.00403000              Text = "ACCESS DENIED!"
00401273  .   FF35 80304000  PUSH DWORD PTR [403080]             hWnd = 0058051A (class='Edit',parent=00C604
00401279  .   E8 FE020000    CALL <JMP.&user32.SetWindowTextA>   SetWindowTextA
0040127E  >   85C0           TEST EAX,EAX
00401280  .v  75 10          JNZ SHORT Crackme6.00401292
00401282  .   68 00304000    PUSH Crackme6.00403000              Text = "ACCESS DENIED!"
00401287  .   FF35 80304000  PUSH DWORD PTR [403080]             hWnd = 0058051A (class='Edit',parent=00C604
0040128D  .   E8 EA020000    CALL <JMP.&user32.SetWindowTextA>   SetWindowTextA
00401292  >   50             PUSH EAX
00401293  .   68 5D304000    PUSH Crackme6.0040305D              ASCII "ACCESS GRAN"
00401298  .   E8 84010000    CALL Crackme6.00401421
0040129D  .   0BC0           OR EAX,EAX
0040129F  .v  75 1F          JNZ SHORT Crackme6.004012C0
004012A1  .   68 0F304000    PUSH Crackme6.0040300F              Text = "ACCESS GRANTED!"
```

حال کلید F8 را بفشارید تا به خط 00401280 :

```
00401261  .   FF75 08        PUSH DWORD PTR [EBP+8]              hWnd
00401264  .   E8 EF020000    CALL <JMP.&user32.GetDlgItemTextA>  GetDlgItemTextA
00401269  .   83F8 0B        CMP EAX,0B
0040126C  .v  72 10          JB SHORT Crackme6.0040127E
0040126E  .   68 00304000    PUSH Crackme6.00403000              Text = "ACCESS DENIED!"
00401273  .   FF35 80304000  PUSH DWORD PTR [403080]             hWnd = 0058051A (class='Edit',parent=00C60438
00401279  .   E8 FE020000    CALL <JMP.&user32.SetWindowTextA>   SetWindowTextA
0040127E  >   85C0           TEST EAX,EAX
00401280  .v  75 10          JNZ SHORT Crackme6.00401292
00401282  .   68 00304000    PUSH Crackme6.00403000              Text = "ACCESS DENIED!"
00401287  .   FF35 80304000  PUSH DWORD PTR [403080]             hWnd = 0058051A (class='Edit',parent=00C60438
0040128D  .   E8 EA020000    CALL <JMP.&user32.SetWindowTextA>   SetWindowTextA
00401292  >   50             PUSH EAX
00401293  .   68 5D304000    PUSH Crackme6.0040305D              ASCII "ACCESS DENI"
00401298  .   E8 84010000    CALL Crackme6.00401421
0040129D  .   0BC0           OR EAX,EAX
0040129F  .v  75 1F          JNZ SHORT Crackme6.004012C0
004012A1  .   68 0F304000    PUSH Crackme6.0040300F              Text = "ACCESS GRANTED!"
004012A6  .   FF35 80304000  PUSH DWORD PTR [403080]             hWnd = 0058051A (class='Edit',parent=00C60438
004012AC  .   E8 CB020000    CALL <JMP.&user32.SetWindowTextA>   SetWindowTextA
004012B1  .   6A 00          PUSH 0                              Enable = FALSE
```

از این پرش (خوشبختانه همواره انجام میشود)هم عبور کنید و در خط 40129F یک دستور JNZ دیگر داریم :

```
0040129D  .  0BC0          OR EAX,EAX
0040129F  .v 75 1F         JNZ SHORT Crackme6.004012C0
004012A1  .  68 0F304000   PUSH Crackme6.0040300F          Text = "ACCESS GRANTED!"
004012A6  .  FF35 80304000 PUSH DWORD PTR [403080]         hWnd = 0058051A (class='Edit',parent=00C60438
004012AC  .  E8 CB020000   CALL <JMP.&user32.SetWindowTextA> SetWindowTextA
004012B1  .  6A 00         PUSH 0                          Enable = FALSE
004012B3  .  FF35 80304000 PUSH DWORD PTR [403080]         hWnd = 0058051A (class='Edit',parent=00C60438
004012B9  .  E8 88020000   CALL <JMP.&user32.EnableWindow> EnableWindow
004012BE  .v EB 10         JMP SHORT Crackme6.004012D0
004012C0  >  68 00304000   PUSH Crackme6.00403000          Text = "ACCESS DENIED!"
004012C5  .  FF35 80304000 PUSH DWORD PTR [403080]         hWnd = 0058051A (class='Edit',parent=00C60438
004012CB  .  E8 AC020000   CALL <JMP.&user32.SetWindowTextA> SetWindowTextA
004012D0  >  6A 0E         PUSH 0E
004012D2  .  68 00304000   PUSH Crackme6.00403000          ASCII "ACCESS DENIED!"
004012D7  .  E8 3F020000   CALL Crackme6.0040151B
004012DC  .  6A 0F         PUSH 0F
004012DE  .  68 0F304000   PUSH Crackme6.0040300F          ASCII "ACCESS GRANTED!"
004012E3  .  E8 33020000   CALL Crackme6.0040151B
004012E8  .v EB 22         JMP SHORT Crackme6.0040130C
```

بسیار خوب،مقدار پرچم(Z) را برابر 1 قرار دهید تا مستقیم به سمت پیام خوب هدایت شویم :

```
0040128D  .  E8 EA020000   CALL <JMP.&user32.SetWindowTextA> SetWindowTextA
00401292  >  50            PUSH EAX
00401293  .  68 5D304000   PUSH Crackme6.0040305D
00401298  .  E8 84010000   CALL Crackme6.00401421
0040129D  .  0BC0          OR EAX,EAX
0040129F  .v 75 1F         JNZ SHORT Crackme6.004012C0
004012A1  .  68 0F304000   PUSH Crackme6.0040300F          Text = "ACCESS GRANTED!"
004012A6  .  FF35 80304000 PUSH DWORD PTR [403080]         hWnd = 0058051A (class='Edit',parent=00C604
004012AC  .  E8 CB020000   CALL <JMP.&user32.SetWindowTextA> SetWindowTextA
004012B1  .  6A 00         PUSH 0                          Enable = FALSE
004012B3  .  FF35 80304000 PUSH DWORD PTR [403080]         hWnd = 0058051A (class='Edit',parent=00C604
004012B9  .  E8 88020000   CALL <JMP.&user32.EnableWindow> EnableWindow
004012BE  .v EB 10         JMP SHORT Crackme6.004012D0
004012C0  >  68 00304000   PUSH Crackme6.00403000          Text = "ACCESS DENIED!"
004012C5  .  FF35 80304000 PUSH DWORD PTR [403080]         hWnd = 0058051A (class='Edit',parent=00C604
004012CB  .  E8 AC0200                                     SetWindowTextA
004012D0  >  6A 0E
004012D2  .  68 003040                                     ASCII "ACCESS DENIED!"
004012D7  .  E8 3F0200
004012DC  .  6A 0F
004012DE  .  68 0F3040                                     ASCII "ACCESS GRANTED!"
004012E3  .  E8 330200
004012E8  .v EB 22
004012EA  >  837D 10 6A
004012EE  .v 75 1C         JNZ SHORT Crackme6.0040130C
004012F0  .  6A 00         PUSH 0                          Result = 0
004012F2  .  FF75 08       PUSH DWORD PTR [EBP+8]          hWnd
004012F5  .  E8 52020000   CALL <JMP.&user32.EndDialog>    EndDialog
004012FA  .v EB 10         JMP SHORT Crackme6.0040130C
004012FC  >  837D 0C 10
```


حال مانند آموز شهای قبلی پچ میکنیم :

401262 JB=JMP
40129F JNZ=Nop

```
0040123C  .  FF35 80304000 PUSH DWORD PTR [403080]         hWnd = 0058051A (class='Edit'
00401242  .  E8 35030000   CALL <JMP.&user32.SetWindowTextA> SetWindowTextA
00401247  .  6A 16         PUSH 16
00401249  .  68 1F304000   PUSH Crackme6.0040301F          ASCII "M>z{ji`}lfah0/Mnk/xnv.
0040124E  .  E8 C8020000   CALL Crackme6.0040151B
00401253  .v E9 B4000000   JMP Crackme6.0040130C
00401258  >  6A 0C         PUSH 0C                         Count = C (12.)
0040125A  .  68 5D304000   PUSH Crackme6.0040305D           Buffer = Crackme6.0040305D
0040125F  .  6A 6B         PUSH 6B                         ControlID = 6B (107.)
00401261  .  FF75 08       PUSH DWORD PTR [EBP+8]          hWnd
00401264  .  E8 EF020000   CALL <JMP.&user32.GetDlgItemTextA> GetDlgItemTextA
00401269  .  83F8 0B       CMP EAX,0B
0040126C  .v EB 10         JMP SHORT Crackme6.0040127E
0040126E  .  68 00304000   PUSH Crackme6.00403000          Text = "NLLJ\\/KJAFJK."
00401273  .  FF35 80304000 PUSH DWORD PTR [403080]         hWnd = 0058051A (class='Edit'
00401279  .  E8 FE020000   CALL <JMP.&user32.SetWindowTextA> SetWindowTextA
0040127E  >  85C0          TEST EAX,EAX
00401280  .v 75 10         JNZ SHORT Crackme6.00401292
00401282  .  68 00304000   PUSH Crackme6.00403000          Text = "NLLJ\\/KJAFJK."
00401287  .  FF35 80304000 PUSH DWORD PTR [403080]         hWnd = 0058051A (class='Edit'
0040128D  .  E8 EA020000   CALL <JMP.&user32.SetWindowTextA> SetWindowTextA
00401292  >  50            PUSH EAX
00401293  .  68 5D304000   PUSH Crackme6.0040305D
00401298  .  E8 84010000   CALL Crackme6.00401421
0040129D  .  0BC0          OR EAX,EAX
0040129F     90            NOP
004012A0     90            NOP
004012A1  .  68 0F304000   PUSH Crackme6.0040300F          Text = "NLLJ\\/HJNA[JK."
004012A6  .  FF35 80304000 PUSH DWORD PTR [403080]         hWnd = 0058051A (class='Edit'
004012AC  .  E8 CB020000   CALL <JMP.&user32.SetWindowTextA> SetWindowTextA
004012B1  .  6A 00         PUSH 0                          Enable = FALSE
```

حال تغییرات را در برنامه ذخیره کنید.