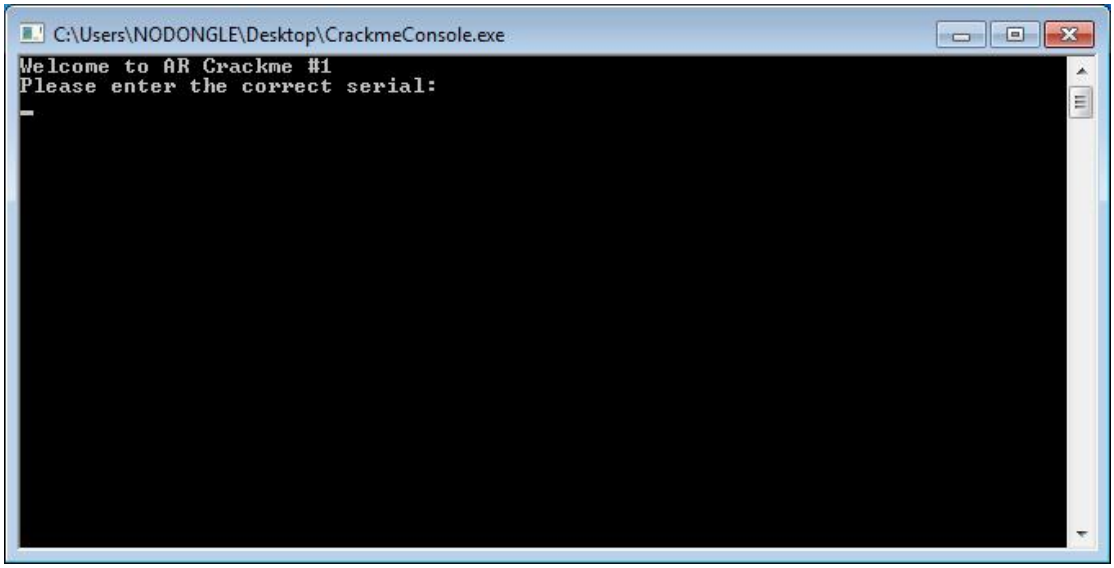
سری آموزش های R4NDOM-فارسی

قسمت ۱۱-فرق بین LAME و NOOB

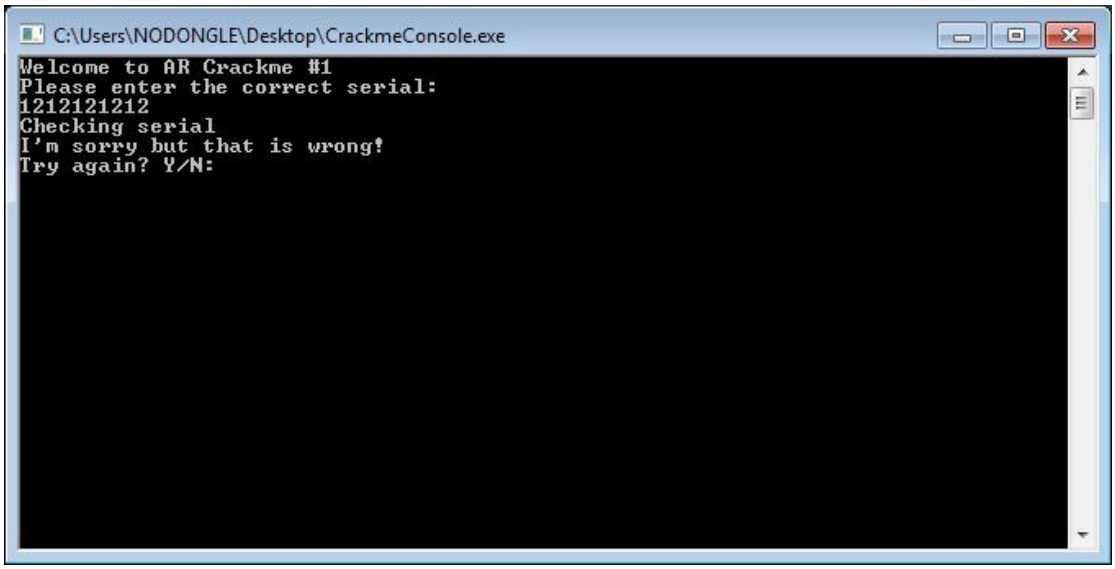
در این قسمت ما دوباره اقدام به پچ میکنیم ولی با این تفاوت که کمی عمیق تر ه مسئله نگاه میکنیم همچنین برنامه ی هدف یک CRACME تحت کنسول است و ما سعی خواهیم کرد که پسورد مخفی شده را پیدا کنیم

بیایید شروع کنیم :

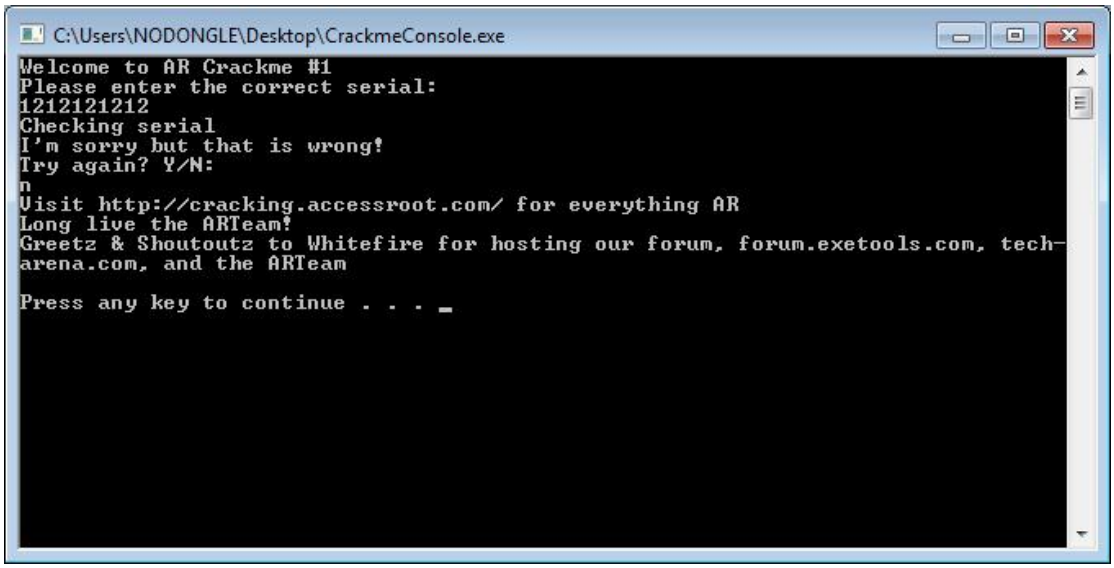
برنامه های کنسول ۳۲ بیتی مانند دیگر برنامه ها هستند با این تفوت که آنها از رابط گرافیکی استفاده نمیکنند ،برنامه هدف (ضمیمه شده) با نام CrackmeConsole.exe است ،برنامه را یک بار اجرا کنید :



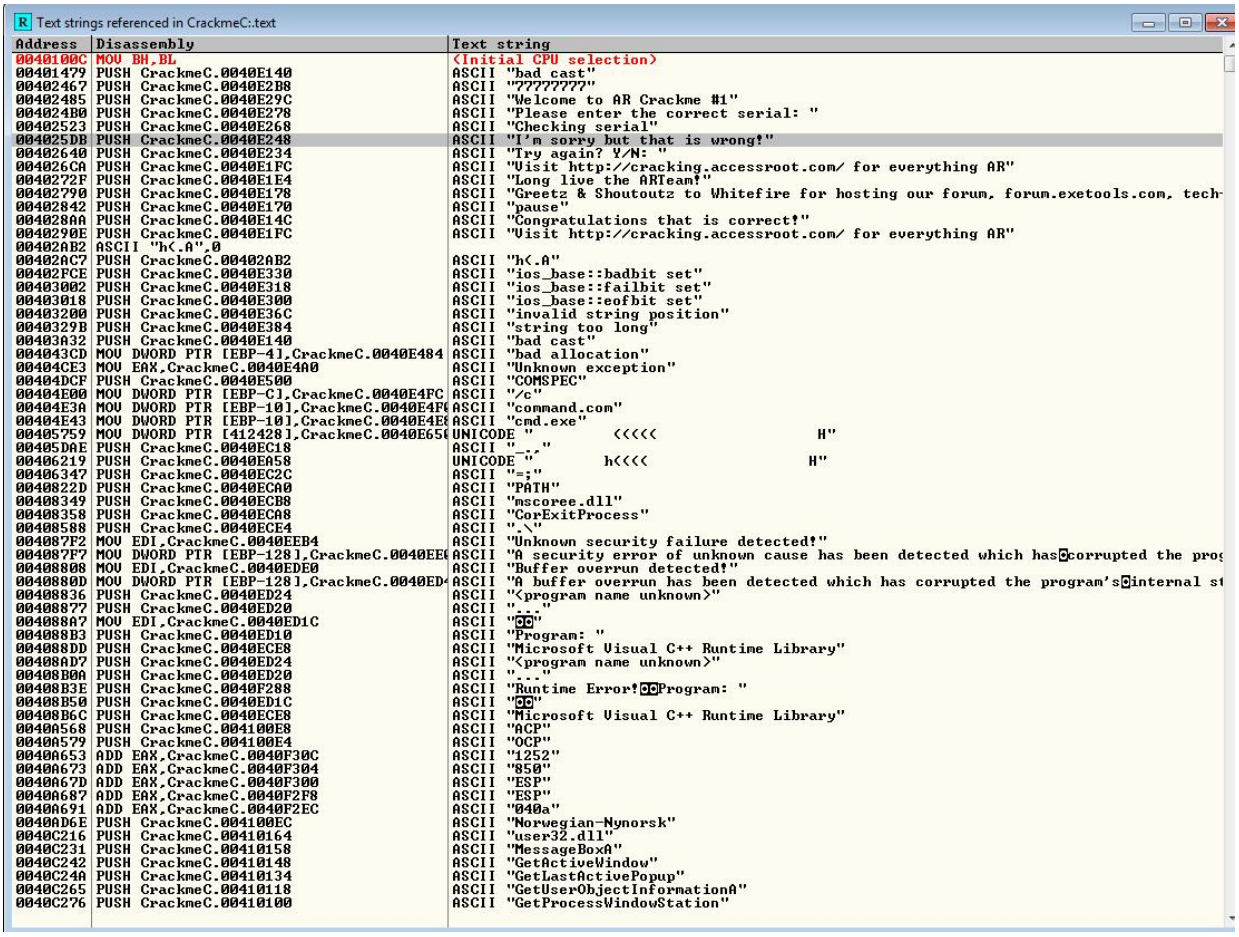
ساده است نه؟یک رمز عبور وارد کنید :



خیلی خب ،کلید "N" را برای خروج بفشارید:



حال برنامه را درون Ollyباز کنید و رشته ها را جستجو کنید:



تا اینجا همه چی خوب بوده؟☺ بسیار خوب بروی پیام هایلایت شده کلیک کنید تا محل نمایش آن برویم :

```
004025BD  .v 74 05          JE SHORT CrackmeC.004025C4
004025BF  .  1BC0           SBB EAX,EAX                          kernel32.BaseThreadInitThunk
004025C1  .  83D8 FF        SBB EAX,-1
004025C4  >  3BC3           CMP EAX,EBX
004025C6  .v 75 13          JNZ SHORT CrackmeC.004025DB
004025C8  >  3BD5           CMP EDX,EBP
004025CA  .v 72 0F          JB SHORT CrackmeC.004025DB
004025CC  .  33C0           XOR EAX,EAX                          kernel32.BaseThreadInitThunk
004025CE  .  3BD5           CMP EDX,EBP
004025D0  .  0F95C0         SETNE AL
004025D3  .  3BC3           CMP EAX,EBX
004025D5  .v 0F84 CF020000  JE CrackmeC.004028AA
004025DB  > 68 48E24000     PUSH CrackmeC.0040E248               ASCII "I'm sorry but that is wr
004025E0  .  68 A02F4100    PUSH CrackmeC.00412FA0
004025E5  .  E8 A6F4FFFF    CALL CrackmeC.00401A90
004025EA  .  83C4 08        ADD ESP,8
004025ED  .  8BF0           MOV ESI,EAX                          kernel32.BaseThreadInitThunk
004025EF  .  6A 0A          PUSH 0A
004025F1  .  8BCE           MOV ECX,ESI
004025F3  .  E8 D8F8FFFF    CALL CrackmeC.00401ED0
004025F8  .  8B0E           MOV ECX,DWORD PTR DS:[ESI]
004025FA  .  8B51 04        MOV EDX,DWORD PTR DS:[ECX+4]
004025FD  .  8A4C32 08      MOV CL,BYTE PTR DS:[EDX+ESI+8]
00402601  .  8D0432         LEA EAX,DWORD PTR DS:[EDX+ESI]
00402604  .  33FF           XOR EDI,EDI
00402606  .  F6C1 06        TEST CL,6
00402609  .v 75 14          JNZ SHORT CrackmeC.0040261F
0040260B  .  8B40 28        MOV EAX,DWORD PTR DS:[EAX+28]
0040260E  .  8B10           MOV EDX,DWORD PTR DS:[EAX]
00402610  .  8BC8           MOV ECX,EAX                          kernel32.BaseThreadInitThunk
00402612  .  FF52 2C        CALL DWORD PTR DS:[EDX+2C]
00402615  .  83F8 FF        CMP EAX,-1
00402618  .v 75 05          JNZ SHORT CrackmeC.0040261F
0040261A  .  BF 04000000    MOV EDI,4
0040261F  >  8B06           MOV EAX,DWORD PTR DS:[ESI]
```

همانطور که میبینید پرشی از آدرس 4025C6 به این قسمت صورت میگیرد(با فلش قرمز رنگ مشخص شده)، و همینطور در خط 4025D5 یک دستورالعمل JE داریم ،بیائید ببینیم به کدام قسمت میرود:

```
004025CE  .  3BD5           CMP EDX,EBP
004025D0  .  0F95C0         SETNE AL
004025D3  .  3BC3           CMP EAX,EBX
004025D5  .v 0F84 CF020000  JE CrackmeC.004028AA
004025DB  > 68 48E24000     PUSH CrackmeC.0040E248               ASCII "I'm sorry but that is wr
004025E0  .  68 A02F4100    PUSH CrackmeC.00412FA0
004025E5  .  E8 A6F4FFFF    CALL CrackmeC.00401A90
004025EA  .  83C4 08        ADD ESP,8
004025ED  .  8BF0           MOV ESI,EAX                          kernel32.BaseThreadInitThunk
004025EF  .  6A 0A          PUSH 0A
004025F1  .  8BCE           MOV ECX,ESI
004025F3  .  E8 D8F8FFFF    CALL CrackmeC.00401ED0
004025F8  .  8B0E           MOV ECX,DWORD PTR DS:[ESI]
004025FA  .  8B51 04        MOV EDX,DWORD PTR DS:[ECX+4]
004025FD  .  8A4C32 08      MOV CL,BYTE PTR DS:[EDX+ESI+8]
00402601  .  8D0432         LEA EAX,DWORD PTR DS:[EDX+ESI]
00402604  .  33FF           XOR EDI,EDI
00402606  .  F6C1 06        TEST CL,6
00402609  .v 75 14          JNZ SHORT CrackmeC.0040261F
0040260B  .  8B40 28        MOV EAX,DWORD PTR DS:[EAX+28]
0040260E  .  8B10           MOV EDX,DWORD PTR DS:[EAX]
00402610  .  8BC8           MOV ECX,EAX                          kernel32.BaseThreadInitThunk
00402612  .  FF52 2C        CALL DWORD PTR DS:[EDX+2C]
00402615  .  83F8 FF        CMP EAX,-1
00402618  .v 75 05          JNZ SHORT CrackmeC.0040261F
0040261A  .  BF 04000000    MOV EDI,4
0040261F  >  8B06           MOV EAX,DWORD PTR DS:[ESI]
00402621  .  8B48 04        MOV ECX,DWORD PTR DS:[EAX+4]
00402624  .  03CE           ADD ECX,ESI
00402626  .  3BFB           CMP EDI,EBX
00402628  .v 74 16          JE SHORT CrackmeC.00402640
0040262A  .  8B41 08        MOV EAX,DWORD PTR DS:[ECX+8]
0040262D  .  8B51 28        MOV EDX,DWORD PTR DS:[ECX+28]
00402630  .  0BC7           OR EAX,EDI
```

پرش نسبتا طولانی است ،همین طور بیایید پائین تر:

```
00402881  .v 72 0D          JB SHORT CrackmeC.00402890
00402883  .  8B5424 24      MOV EDX,DWORD PTR SS:[ESP+24]
00402887  .  52             PUSH EDX                             CrackmeC.<ModuleEntryPoint>
00402888  .  E8 C01E0000    CALL CrackmeC.0040474D
0040288D  .  83C4 04        ADD ESP,4
00402890  >  8B4C24 40      MOV ECX,DWORD PTR SS:[ESP+40]
00402894  .  64:890D 00000000 MOV DWORD PTR FS:[0],ECX
0040289B  .  8B4C24 3C      MOV ECX,DWORD PTR SS:[ESP+3C]        ntdll.77D2E115
0040289F  .  33C0           XOR EAX,EAX                          kernel32.BaseThreadInitThunk
004028A1  .  E8 C0260000    CALL CrackmeC.00404F66
004028A6  .  83C4 4C        ADD ESP,4C
004028A9  .  C3             RETN
004028AA  > 68 4CE14000     PUSH CrackmeC.0040E14C               ASCII "Congratulations that is correc
004028AF  .  68 A02F4100    PUSH CrackmeC.00412FA0
004028B4  .  E8 D7F1FFFF    CALL CrackmeC.00401A90
004028B9  .  83C4 08        ADD ESP,8
004028BC  .  8BF0           MOV ESI,EAX                          kernel32.BaseThreadInitThunk
004028BE  .  6A 0A          PUSH 0A
004028C0  .  8BCE           MOV ECX,ESI
004028C2  .  E8 09F6FFFF    CALL CrackmeC.00401ED0
004028C7  .  8B06           MOV EAX,DWORD PTR DS:[ESI]
004028C9  .  8B48 04        MOV ECX,DWORD PTR DS:[EAX+4]
004028CC  .  8D0431         LEA EAX,DWORD PTR DS:[ECX+ESI]
004028CF  .  8A48 08        MOV CL,BYTE PTR DS:[EAX+8]
004028D2  .  33FF           XOR EDI,EDI
004028D4  .  F6C1 06        TEST CL,6
004028D7  .v 75 14          JNZ SHORT CrackmeC.004028ED
004028D9  .  8B40 28        MOV EAX,DWORD PTR DS:[EAX+28]
```

اها☺ ،به نظر اینجا باید فرود بیاییم ،ولی بیائید برگردیم :

```
00402571  .   8B41 08        MOV EAX,DWORD PTR DS:[ECX+8]
00402574  .   8B51 28        MOV EDX,DWORD PTR DS:[ECX+28]
00402577  .   0BC7           OR EAX,EDI
00402579  .   3BD3           CMP EDX,EBX
0040257B  .v  75 08          JNZ SHORT CrackmeC.00402580
0040257D  .   83C8 04        OR EAX,4
00402580  >   53             PUSH EBX
00402581  .   50             PUSH EAX                                kernel32.BaseThreadInitThunk
00402582  .   E8 100A0000    CALL CrackmeC.00402F97
00402587  >   837C24 2C 10   CMP DWORD PTR SS:[ESP+2C],10
0040258C  .   8B7C24 18      MOV EDI,DWORD PTR SS:[ESP+18]
00402590  .v  73 04          JNB SHORT CrackmeC.00402596
00402592  .   8D7C24 18      LEA EDI,DWORD PTR SS:[ESP+18]
00402596  >   8B5424 44      MOV EDX,DWORD PTR SS:[ESP+44]
0040259A  .   3BD3           CMP EDX,EBX
0040259C  .   8B6C24 28      MOV EBP,DWORD PTR SS:[ESP+28]
004025A0  .v  74 26          JE SHORT CrackmeC.004025C8
004025A2  .   3BD5           CMP EDX,EBP
004025A4  .   8BCA           MOV ECX,EDX                             CrackmeC.<ModuleEntryPoint>
004025A6  .v  72 02          JB SHORT CrackmeC.004025AA
004025A8  .   8BCD           MOV ECX,EBP
004025AA  >   837C24 48 10   CMP DWORD PTR SS:[ESP+48],10
004025AF  .   8B7424 34      MOV ESI,DWORD PTR SS:[ESP+34]
004025B3  .v  73 04          JNB SHORT CrackmeC.004025B9
004025B5  .   8D7424 34      LEA ESI,DWORD PTR SS:[ESP+34]
004025B9  >   33C0           XOR EAX,EAX                             kernel32.BaseThreadInitThunk
004025BB  .   F3:A6          REPE CMPS BYTE PTR ES:[EDI],BYTE PTR DS:
004025BD  .v  74 05          JE SHORT CrackmeC.004025C4
004025BF  .   1BC0           SBB EAX,EAX                             kernel32.BaseThreadInitThunk
004025C1  .   83D8 FF        SBB EAX,-1
004025C4  >   3BC3           CMP EAX,EBX
004025C6  .v  75 13          JNZ SHORT CrackmeC.004025DB
004025C8  >   3BD5           CMP EDX,EBP
004025CA  .v  72 0F          JB SHORT CrackmeC.004025DB
004025CC  .   33C0           XOR EAX,EAX                             kernel32.BaseThreadInitThunk
004025CE  .   3BD5           CMP EDX,EBP
004025D0  .   0F95C0         SETNE AL
004025D3  .   3BC3           CMP EAX,EBX
004025D5  .v  0F84 CF020000  JE CrackmeC.004028AA
004025DB  >   68 48E24000    PUSH CrackmeC.0040E248                  ASCII "I'm sorry but that is wrong!"
004025E0  .   68 A02F4100    PUSH CrackmeC.00412FA0
004025E5  .   E8 A6F4FFFF    CALL CrackmeC.00401A90
004025EA  .   83C4 08        ADD ESP,8                               kernel32.BaseThreadInitThunk
004025ED  .   8BF0           MOV ESI,EAX
004025EF  .   6A 0A          PUSH 0A
004025F1  .   8BCE           MOV ECX,ESI
004025F3  .   E8 D8F8FFFF    CALL CrackmeC.00401ED0
004025F8  .   8B0E           MOV ECX,DWORD PTR DS:[ESI]
004025FA  .   8B51 04        MOV EDX,DWORD PTR DS:[ECX+4]
004025FD  .   8A4C32 08      MOV CL,BYTE PTR DS:[EDX+ESI+8]
00402601  .   8D0432         LEA EAX,DWORD PTR DS:[EDX+ESI]
00402604  .   33FF           XOR EDI,EDI
00402606  .   F6C1 06        TEST CL,6
00402609  .v  75 14          JNZ SHORT CrackmeC.0040261F
0040260B  .   8B40 28        MOV EAX,DWORD PTR DS:[EAX+28]
0040260E  .   8B10           MOV EDX,DWORD PTR DS:[EAX]
```

مشخص شد که این پرش از آدرس 4025D5 به قسمت Good Boyمیرود،کمی بالاتر در ادرس 4025CA یک دستور JB (پرشی) میبینید که آن هم به سمت Bad Boy میرود :

```
004025BF  .   1BC0           SBB EAX,EAX                             kernel32.BaseThreadInitThunk
004025C1  .   83D8 FF        SBB EAX,-1
004025C4  >   3BC3           CMP EAX,EBX
004025C6  .v  75 13          JNZ SHORT CrackmeC.004025DB
004025C8  >   3BD5           CMP EDX,EBP
004025CA  .v  72 0F          JB SHORT CrackmeC.004025DB
004025CC  .   33C0           XOR EAX,EAX                             kernel32.BaseThreadInitThunk
004025CE  .   3BD5           CMP EDX,EBP
004025D0  .   0F95C0         SETNE AL
004025D3  .   3BC3           CMP EAX,EBX
004025D5  .v  0F84 CF020000  JE CrackmeC.004028AA
004025DB  >   68 48E24000    PUSH CrackmeC.0040E248                  ASCII "I'm sorry but that is wrong!"
004025E0  .   68 A02F4100    PUSH CrackmeC.00412FA0
004025E5  .   E8 A6F4FFFF    CALL CrackmeC.00401A90
004025EA  .   83C4 08        ADD ESP,8
```

و همینطور در آدرس 4025C6 یک پرش به سمت Bad Boy داریم :

```
004025BB  .   F3:A6          REPE CMPS BYTE PTR ES:[EDI],BYTE PTR DS:
004025BD  .v  74 05          JE SHORT CrackmeC.004025C4
004025BF  .   1BC0           SBB EAX,EAX                             kernel32.BaseThreadInitThunk
004025C1  .   83D8 FF        SBB EAX,-1
004025C4  >   3BC3           CMP EAX,EBX
004025C6  .v  75 13          JNZ SHORT CrackmeC.004025DB
004025C8  >   3BD5           CMP EDX,EBP
004025CA  .v  72 0F          JB SHORT CrackmeC.004025DB
004025CC  .   33C0           XOR EAX,EAX                             kernel32.BaseThreadInitThunk
004025CE  .   3BD5           CMP EDX,EBP
004025D0  .   0F95C0         SETNE AL
004025D3  .   3BC3           CMP EAX,EBX
004025D5  .v  0F84 CF020000  JE CrackmeC.004028AA
004025DB  >   68 48E24000    PUSH CrackmeC.0040E248                  ASCII "I'm sorry but that is wrong!"
004025E0  .   68 A02F4100    PUSH CrackmeC.00412FA0
004025E5  .   E8 A6F4FFFF    CALL CrackmeC.00401A90
004025EA  .   83C4 08        ADD ESP,8
```

همینطور که مشخص است تنها پرشی که به سمت Good Boy میـرود در آدرس 4025D5 اسـت،خب سـناریو مـشخص شـد ،تمـام پرش ها به سمت Bad Boy نباید ارجاع شود،تا به آدرس 4025D5 و از آنجا مستقیم به Good Boyبرویم ،اگر کمـی بـالاتر را نگـاه کنیم درست در آدرس 402582 ما اولین دستورالعمل call / compare را میبینیم :

```
00402579  .  3BD3          CMP EDX,EBX
0040257B  .v 75 03         JNZ SHORT CrackmeC.00402580
0040257D  .  83C8 04       OR EAX,4
00402580  >  53            PUSH EBX
00402581  .  50            PUSH EAX                                 kernel32.BaseThreadInitThunk
00402582  .  E8 100A0000   CALL CrackmeC.00402F97
00402587  >  837C24 2C 10  CMP DWORD PTR SS:[ESP+2C],10
0040258C  .  8B7C24 18     MOV EDI,DWORD PTR SS:[ESP+18]
00402590  .v 73 04         JNB SHORT CrackmeC.00402596
00402592  .  8D7C24 18     LEA EDI,DWORD PTR SS:[ESP+18]
00402596  >  8B5424 44     MOV EDX,DWORD PTR SS:[ESP+44]
0040259A  .  3BD3          CMP EDX,EBX
0040259C  .  8B6C24 28     MOV EBP,DWORD PTR SS:[ESP+28]
004025A0  .v 74 26         JE SHORT CrackmeC.004025C8
004025A2  .  3BD5          CMP EDX,EBP
004025A4  .  8BCA          MOV ECX,EDX                              CrackmeC.<ModuleEntryPoint>
004025A6  .v 72 02         JB SHORT CrackmeC.004025AA
004025A8  .  8BCD          MOV ECX,EBP
004025AA  >  837C24 48 10  CMP DWORD PTR SS:[ESP+48],10
004025AF  .  8B7424 34     MOV ESI,DWORD PTR SS:[ESP+34]
004025B3  .v 73 04         JNB SHORT CrackmeC.004025B9
004025B5  .  8D7424 34     LEA ESI,DWORD PTR SS:[ESP+34]
004025B9  >  33C0          XOR EAX,EAX                              kernel32.BaseThreadInitThunk
004025BB  .  F3:A6         REPE CMPS BYTE PTR ES:[EDI],BYTE PTR DS:
```

اگر باز کمی به بالا نگاه کنیم یک پرش در آدرس 40256F میبینیم که از این فراخوانی عبور میکند :

```
0040255F  .v 75 05         JNZ SHORT CrackmeC.00402566
00402561  .  BF 04000000   MOV EDI,4
00402566  >  8B06          MOV EAX,DWORD PTR DS:[ESI]
00402568  .  8B48 04       MOV ECX,DWORD PTR DS:[EAX+4]
0040256B  .  03CE          ADD ECX,ESI
0040256D  .  3BFB          CMP EDI,EBX
0040256F  .v 74 16         JE SHORT CrackmeC.00402587
00402571  .  8B41 08       MOV EAX,DWORD PTR DS:[ECX+8]
00402574  .  8B51 28       MOV EDX,DWORD PTR DS:[ECX+28]
00402577  .  0BC7          OR EAX,EDI
00402579  .  3BD3          CMP EDX,EBX
0040257B  .v 75 09         JNZ SHORT CrackmeC.00402580
0040257D  .  83C8 04       OR EAX,4
00402580  >  53            PUSH EBX
00402581  .  50            PUSH EAX                                 kernel32.BaseThreadInitThunk
00402582  .  E8 100A0000   CALL CrackmeC.00402F97
00402587  >  837C24 2C 10  CMP DWORD PTR SS:[ESP+2C],10
0040258C  .  8B7C24 18     MOV EDI,DWORD PTR SS:[ESP+18]
00402590  .v 73 04         JNB SHORT CrackmeC.00402596
00402592  .  8D7C24 18     LEA EDI,DWORD PTR SS:[ESP+18]
00402596  >  8B5424 44     MOV EDX,DWORD PTR SS:[ESP+44]
0040259A  .  3BD3          CMP EDX,EBX
0040259C  .  8B6C24 28     MOV EBP,DWORD PTR SS:[ESP+28]
004025A0  .v 74 26         JE SHORT CrackmeC.004025C8
004025A2  .  3BD5          CMP EDX,EBP
```

Jump past the call

باز کمی بالاتر یک فراخوانی و مقایسه دیگر وجود دارد، حالا بروی هر دو تابع BP میگذاریم :

```
00402542  .  8B42 04       MOV EAX,DWORD PTR DS:[EDX+4]
00402545  .  8A4C30 08     MOV CL,BYTE PTR DS:[EAX+ESI+8]
00402549  .  03C6          ADD EAX,ESI
0040254B  .  33FF          XOR EDI,EDI
0040254D  .  F6C1 06       TEST CL,6
00402550  .v 75 14         JNZ SHORT CrackmeC.00402566
00402552  .  8B40 28       MOV EAX,DWORD PTR DS:[EAX+28]
00402555  .  8B10          MOV EDX,DWORD PTR DS:[EAX]
00402557  .  8BC8          MOV ECX,EAX                              kernel32.BaseThreadInitThunk
00402559  .  FF52 2C       CALL DWORD PTR DS:[EDX+2C]
0040255C  .  83F8 FF       CMP EAX,-1
0040255F  .v 75 05         JNZ SHORT CrackmeC.00402566
00402561  .  BF 04000000   MOV EDI,4
00402566  >  8B06          MOV EAX,DWORD PTR DS:[ESI]
00402568  .  8B48 04       MOV ECX,DWORD PTR DS:[EAX+4]
0040256B  .  03CE          ADD ECX,ESI
0040256D  .  3BFB          CMP EDI,EBX
0040256F  .v 74 16         JE SHORT CrackmeC.00402587
00402571  .  8B41 08       MOV EAX,DWORD PTR DS:[ECX+8]
00402574  .  8B51 28       MOV EDX,DWORD PTR DS:[ECX+28]
00402577  .  0BC7          OR EAX,EDI
00402579  .  3BD3          CMP EDX,EBX
0040257B  .v 75 09         JNZ SHORT CrackmeC.00402580
0040257D  .  83C8 04       OR EAX,4
00402580  >  53            PUSH EBX
00402581  .  50            PUSH EAX                                 kernel32.BaseThreadInitThunk
00402582  .  E8 100A0000   CALL CrackmeC.00402F97
00402587  >  837C24 2C 10  CMP DWORD PTR SS:[ESP+2C],10
0040258C  .  8B7C24 18     MOV EDI,DWORD PTR SS:[ESP+18]
00402590  .v 73 04         JNB SHORT CrackmeC.00402596
00402592  .  8D7C24 18     LEA EDI,DWORD PTR SS:[ESP+18]
00402596  >  8B5424 44     MOV EDX,DWORD PTR SS:[ESP+44]
0040259A  .  3BD3          CMP EDX,EBX
0040259C  .  8B6C24 28     MOV EBP,DWORD PTR SS:[ESP+28]
```

حالا برنامه را درون Olly اجرا کرده و یک پسورد دلخواه وارد کـرده مـن "12121212 " را وارد میکـنم و olly بـروی تـابع اولـی متوقـف میشود:

```
00402549  .  03C6          ADD EAX,ESI                                CrackmeC.00412FA0
0040254B  .  33FF          XOR EDI,EDI
0040254D  .  F6C1 06       TEST CL,6
00402550  .v 75 14         JNZ SHORT CrackmeC.00402566
00402552  .  8B40 28       MOV EAX,DWORD PTR DS:[EAX+28]
00402555  .  8B10          MOV EDX,DWORD PTR DS:[EAX]                 CrackmeC.0040E408
00402557  .  8BC8          MOV ECX,EAX                                CrackmeC.00412F40
00402559  .  FF52 2C       CALL DWORD PTR DS:[EDX+2C]                 CrackmeC.004037A1
0040255C  .  83F8 FF       CMP EAX,-1
0040255F  .v 75 05         JNZ SHORT CrackmeC.00402566
00402561  .  BF 04000000   MOV EDI,4
00402566  >  8B06          MOV EAX,DWORD PTR DS:[ESI]                 CrackmeC.0040E470
00402568  .  8B48 04       MOV ECX,DWORD PTR DS:[EAX+4]
0040256B  .  03CE          ADD ECX,ESI                                CrackmeC.00412FA0
0040256D  .  3BFB          CMP EDI,EBX
0040256F  .v 74 16         JE SHORT CrackmeC.00402587
00402571  .  8B41 08       MOV EAX,DWORD PTR DS:[ECX+8]
00402574  .  8B51 28       MOV EDX,DWORD PTR DS:[ECX+28]
00402577  .  0BC7          OR EAX,EDI
00402579  .  3BD3          CMP EDX,EBX
0040257B  .v 75 03         JNZ SHORT CrackmeC.00402580
0040257D  .  83C8 04       OR EAX,4
00402580  >  53            PUSH EBX
00402581  .  50            PUSH EAX                                   CrackmeC.00412F40
00402582  .  E8 100A0000   CALL CrackmeC.00402F97
00402587  >  837C24 2C 10  CMP DWORD PTR SS:[ESP+2C],10
0040258C  .  8B7C24 18     MOV EDI,DWORD PTR SS:[ESP+18]
00402590  .v 73 04         JNB SHORT CrackmeC.00402596
00402592  .  8D7C24 18     LEA EDI,DWORD PTR SS:[ESP+18]
00402596  >  8B5424 44     MOV EDX,DWORD PTR SS:[ESP+44]
0040259A  .  3BD3          CMP EDX,EBX
0040259C  .  8B6C24 28     MOV EBP,DWORD PTR SS:[ESP+28]
```

حال کلید F8 را بفشارید تا به آدرس 40256F برسید،همانطور که میبینید این پرش از فراخوانی و مقایسه عبور می کند :

```
0040254D  .  F6C1 06       TEST CL,6
00402550  .v 75 14         JNZ SHORT CrackmeC.00402566
00402552  .  8B40 28       MOV EAX,DWORD PTR DS:[EAX+28]              CrackmeC.00404D25
00402555  .  8B10          MOV EDX,DWORD PTR DS:[EAX]
00402557  .  8BC8          MOV ECX,EAX                                CrackmeC.0040E470
00402559  .  FF52 2C       CALL DWORD PTR DS:[EDX+2C]                 CrackmeC.004037A1
0040255C  .  83F8 FF       CMP EAX,-1
0040255F  .v 75 05         JNZ SHORT CrackmeC.00402566
00402561  .  BF 04000000   MOV EDI,4
00402566  >  8B06          MOV EAX,DWORD PTR DS:[ESI]                 CrackmeC.0040E470
00402568  .  8B48 04       MOV ECX,DWORD PTR DS:[EAX+4]
0040256B  .  03CE          ADD ECX,ESI                                CrackmeC.00412FA0
0040256D  .  3BFB          CMP EDI,EBX
0040256F  .v 74 16         JE SHORT CrackmeC.00402587
00402571  .  8B41 08       MOV EAX,DWORD PTR DS:[ECX+8]
00402574  .  8B51 28       MOV EDX,DWORD PTR DS:[ECX+28]              CrackmeC.00412F40
00402577  .  0BC7          OR EAX,EDI
00402579  .  3BD3          CMP EDX,EBX
0040257B  .v 75 03         JNZ SHORT CrackmeC.00402580
0040257D  .  83C8 04       OR EAX,4
00402580  >  53            PUSH EBX
00402581  .  50            PUSH EAX                                   CrackmeC.0040E470
00402582  .  E8 100A0000   CALL CrackmeC.00402F97
00402587  >  837C24 2C 10  CMP DWORD PTR SS:[ESP+2C],10
0040258C  .  8B7C24 18     MOV EDI,DWORD PTR SS:[ESP+18]
00402590  .v 73 04         JNB SHORT CrackmeC.00402596
00402592  .  8D7C24 18     LEA EDI,DWORD PTR SS:[ESP+18]
00402596  >  8B5424 44     MOV EDX,DWORD PTR SS:[ESP+44]
0040259A  .  3BD3          CMP EDX,EBX
```

Jumps past our call

کلید F8را آنقدر بفشارید تا به آدرس 4025C6 برسید ،این همان آدرسی است که ما را به سمت Bad Boy میبرد:

```
004025A6  .v 72 02         JB SHORT CrackmeC.004025AA
004025A8  .  8BCD          MOV ECX,EBP
004025AA  >  837C24 48 10  CMP DWORD PTR SS:[ESP+48],10
004025AF  .  8B7424 34     MOV ESI,DWORD PTR SS:[ESP+34]
004025B3  .v 73 04         JNB SHORT CrackmeC.004025B9
004025B5  .  8D7424 34     LEA ESI,DWORD PTR SS:[ESP+34]
004025B9  >  33C0          XOR EAX,EAX
004025BB  .  F3:A6         REPE CMPS BYTE PTR ES:[EDI],BYTE PTR DS:
004025BD  .v 74 05         JE SHORT CrackmeC.004025C4
004025BF  .  1BC0          SBB EAX,EAX
004025C1  .  83D8 FF       SBB EAX,-1
004025C4  >  3BC3          CMP EAX,EBX
004025C6  .v 75 13         JNZ SHORT CrackmeC.004025DB
004025C8  >  3BD5          CMP EDX,EBP
004025CA  .v 72 0F         JB SHORT CrackmeC.004025DB
004025CC  .  33C0          XOR EAX,EAX
004025CE  .  3BD5          CMP EDX,EBP
004025D0  .  0F95C0        SETNE AL
004025D3  .  3BC3          CMP EAX,EBX
004025D5  .v 0F84 CF020000 JE CrackmeC.004028AA
004025DB  >  68 48E24000   PUSH CrackmeC.0040E248                     ASCII "I'm sorry but that is wrong!"
004025E0  .  68 A02F4100   PUSH CrackmeC.00412FA0
004025E5  .  E8 A6F4FFFF   CALL CrackmeC.00401A90
004025EA  .  83C4 08       ADD ESP,8
004025ED  .  8BF0          MOV ESI,EAX
004025EF  .  6A 0A         PUSH 0A
004025F1  .  8BCE          MOV ECX,ESI
004025F3  .  E8 D8F8FFFF   CALL CrackmeC.00401ED0
004025F8  .  8B0E          MOV ECX,DWORD PTR DS:[ESI]
```

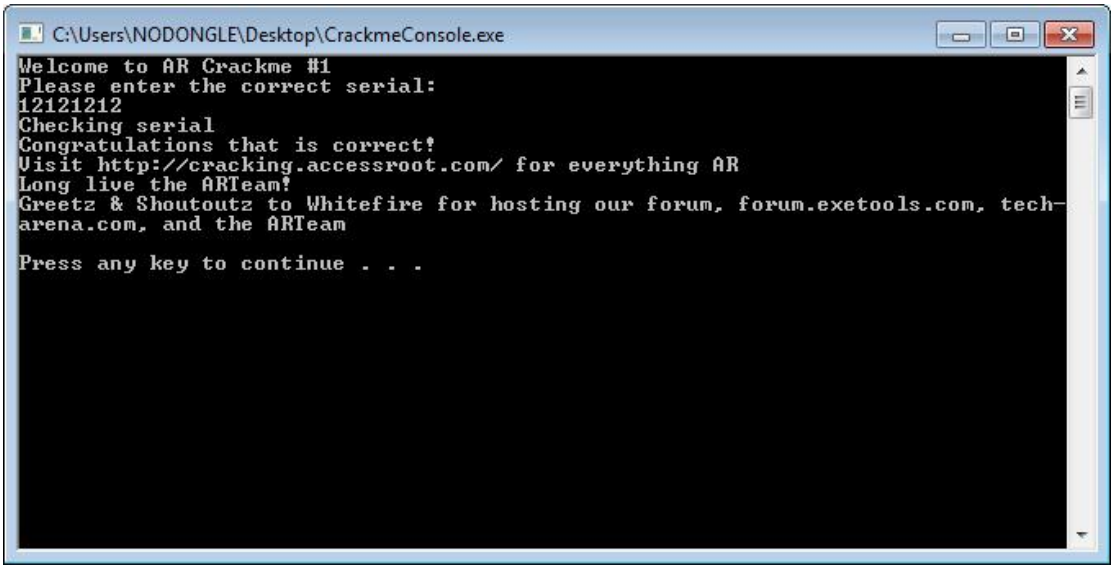
Jumps to bad boy

حالا با تغییر پرچم Z=1 این پرش را غیر فعال کنید :

```
C 0   ES 0023
P 1   CS 001B
A 0   SS 0023
Z 1   DS 0023
S 1   FS 003B
T 0   GS 0000
D 0
O 0   LastErr
```

حالا کلید F8را بفشارید،میبینید که مستقیم به سمت Good Boy ارجاع داده شدیم :

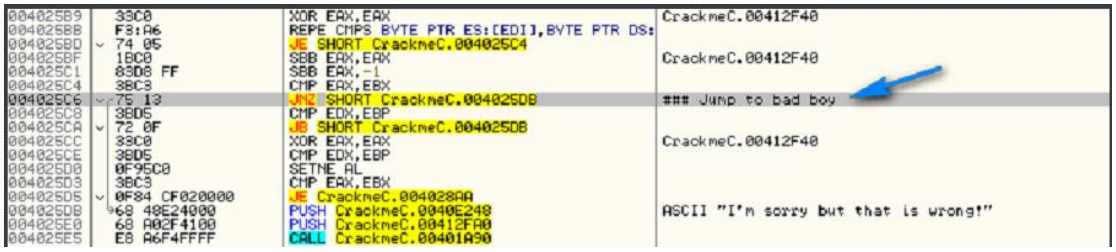
حالا به برنامه نگاه کنید :



حالا طبق آموزش های قبلی پرش قبلی را با دستور Nop تغییر و نتیجه را ثبت و برنامه را اجرا کنید،و یک پسورد وارد کنید اما اینبار طول پسورد کوتاه یا بلند باشد ☺ برنامه پیام Bad Boy را نمایش داد ؟☺ بیایید علت ماجرا را بفهمیم :

حفاری عمیق تر

درواقع اگر بخواهیم از نگاه آموزش قبلی به این موضوع نگاه کنیم درواقع ما از تکنیک LAME استفاده کرده ایم بیایید بروی پرشی که ما را به Bad Boy ارجاع میدهد (که قبلا پچ کردیم) متمرکز شویم و بفهمیم که چرا با وجود پچ کردن باز هم با تغییر طول پسورد این پچ درست عمل نمیکند ،از طرفی با درج توضیحات میتوان برنامه را مستند سازی کرد ،من از ## در ابتدای توضیحات استفاده میکنم که موقع رجوع پیدا کردنش راحت تر باشد،البته شما میتوانید هر جور که راحتید عمل کنید:



به قسمت انتخاب شده (کادر قرمز رنگ) تصویر زیر نگاه کنید:

```
004025AA   837C24 48 10    CMP DWORD PTR SS:[ESP+48],10
004025AF   8B7424 34       MOV ESI,DWORD PTR SS:[ESP+34]
004025B3 v 73 04           JNB SHORT CrackmeC.004025B9
004025B5   8D7424 34       LEA ESI,DWORD PTR SS:[ESP+34]
004025B9   33C0            XOR EAX,EAX                                  CrackmeC.00412F40
004025BB   F3:A6           REPE CMPS BYTE PTR ES:[EDI],BYTE PTR DS:
004025BD v 74 05           JE SHORT CrackmeC.004025C4
004025BF   1BC0            SBB EAX,EAX                                  CrackmeC.00412F40
004025C1   83D8 FF         SBB EAX,-1
004025C4   3BC3            CMP EAX,EBX
004025C6 v 75 13           JNZ SHORT CrackmeC.004025DB                  ### Jump to bad boy
004025C8   3BD5            CMP EDX,EBP
004025CA v 72 0F           JB SHORT CrackmeC.004025DB
004025CC   33C0            XOR EAX,EAX                                  CrackmeC.00412F40
004025CE   3BD5            CMP EDX,EBP
004025D0   0F95C0          SETNE AL
004025D3   3BC3            CMP EAX,EBX
004025D5 v 0F84 CF020000   JE CrackmeC.004028AA
004025DB   68 48E24000     PUSH CrackmeC.0040E248                       ASCII "I'm sorry but that is wrong!"
004025E0   68 A02F4100     PUSH CrackmeC.00412FA0
```

چند دستور SBB و یک دستور مقایسه میبینید که در واقع معنی خاصی برای درک کلی کد به ما نمیدهند

دستورالعمل sbb (subtract with borrow) مشابه دستور sub است با این تفاوت که حاصل تفریق عملوند دوم و CF از عملوند اول را محاسبه می نماید(dest:=dest-src-CF)

اما اگر به تص.یر زیر نگاه کنید همه چیز مشخص خواهد شد:

```
004025A2   3BD5            CMP EDX,EBP
004025A4   8BCA            MOV ECX,EDX                                  CrackmeC.0040E408
004025A6 v 72 02           JB SHORT CrackmeC.004025AA
004025A8   8BCD            MOV ECX,EBP
004025AA   837C24 48 10    CMP DWORD PTR SS:[ESP+48],10
004025AF   8B7424 34       MOV ESI,DWORD PTR SS:[ESP+34]
004025B3 v 73 04           JNB SHORT CrackmeC.004025B9
004025B5   8D7424 34       LEA ESI,DWORD PTR SS:[ESP+34]
004025B9   33C0            XOR EAX,EAX                                  CrackmeC.00412F40
004025BB   F3:A6           REPE CMPS BYTE PTR ES:[EDI],BYTE PTR DS:
004025BD v 74 05           JE SHORT CrackmeC.004025C4
004025BF   1BC0            SBB EAX,EAX                                  CrackmeC.00412F40
004025C1   83D8 FF         SBB EAX,-1
004025C4   3BC3            CMP EAX,EBX
004025C6 v 75 13           JNZ SHORT CrackmeC.004025DB                  ### Jump to bad boy
004025C8   3BD5            CMP EDX,EBP
004025CA v 72 0F           JB SHORT CrackmeC.004025DB
004025CC   33C0            XOR EAX,EAX                                  CrackmeC.00412F40
004025CE   3BD5            CMP EDX,EBP
004025D0   0F95C0          SETNE AL
004025D3   3BC3            CMP EAX,EBX
004025D5 v 0F84 CF020000   JE CrackmeC.004028AA
004025DB   68 48E24000     PUSH CrackmeC.0040E248                       ASCII "I'm sorry but that is wrong!"
004025E0   68 A02F4100     PUSH CrackmeC.00412FA0
004025E5   E8 A6F4FFFF     CALL CrackmeC.00401A90
```

دستورالعمل Lea : این دستور آدرس موثر یک محل خاص از حافظه را درون یک ثبات همه منظوره ذخیره می کند. منظور از آدرس موثر آدرس نهائی حافظه بعد از کلیه محاسبات آدرسی است. همچنین این دستور بروی پرچم ها تاثیری ندارد

همین طور خط بعدی دستور العمل REPE CMPS را داریم :

Intel x86 Instructions

File Edit Bookmark Options Help

Contents Index Back Print

REP/REPE/REPZ/REPNE/REPNZ—Repeat String Operation Prefix

See also

F2 AF	REPNE SCAS *m32*	Find EAX, starting at ES:[(E)DI]

Description

Repeats a string instruction the number of times specified in the count register ((E)CX) or until the indicated condition of the ZF flag is no longer met. The REP (repeat), REPE (repeat while equal), REPNE (repeat while not equal), REPZ (repeat while zero), and REPNZ (repeat while not zero) mnemonics are prefixes that can be added to one of the string instructions. The REP prefix can be added to the INS, OUTS, MOVS, LODS, and STOS instructions, and the REPE, REPNE, REPZ, and REPNZ prefixes can be added to the CMPS and SCAS instructions. (The REPZ and REPNZ prefixes are synonymous forms of the REPE and REPNE prefixes, respectively.) The behavior of the REP prefix is undefined when used with non–string instructions.

The REP prefixes apply only to one string instruction at a time. To repeat a block of instructions, use the LOOP instruction or another looping construct.

All of these repeat prefixes cause the associated instruction to be repeated until the count in register (E)CX is decremented to 0 (see the following table). (If the current address–size attribute is 32, register ECX is used as a counter, and if the address–size attribute is 16, the CX register is used.) The REPE, REPNE, REPZ, and REPNZ prefixes also check the state of the ZF flag after each iteration and terminate the repeat loop if the ZF flag is not in the specified state. When both termination conditions are tested, the cause of a repeat termination can be determined either by testing the (E)CX register with a JECXZ instruction or by testing the ZF flag with a JZ, JNZ, and JNE instruction.

Repeat Conditions

Repeat Prefix	Termination Condition 1	Termination Condition 2
REP	ECX=0	None
REPE/REPZ	ECX=0	ZF=0
REPNE/REPNZ	ECX=0	ZF=1

When the REPE/REPZ and REPNE/REPNZ prefixes are used, the ZF flag does not require initialization because both the CMPS and SCAS instructions affect the ZF flag according to the results of the comparisons they make.

A repeating string operation can be suspended by an exception or interrupt. When this happens, the state of the registers is preserved to allow the string operation to be resumed upon a return from the exception or interrupt handler. The source and destination registers point to the next string elements to be operated on, the EIP register points to the string instruction, and the ECX register has the value it held following the last successful iteration of the instruction. This mechanism allows long string operations to proceed without affecting the interrupt response time of the system.

When a fault occurs during the execution of a CMPS or SCAS instruction that is prefixed with REPE or REPNE, the EFLAGS value is restored to the state prior to the execution of the instruction. Since the SCAS and CMPS instructions do not use EFLAGS as an input, the processor can resume the instruction after the page fault handler.

Use the REP INS and REP OUTS instructions with caution. Not all I/O ports can handle the rate at which these instructions execute.

A REP STOS instruction is the fastest way to initialize a large block of memory.

دستور بزرگی است نه؟☺ اما کارکرد بسیار جالب و در عین حال ساده ایی دارد،دستور REPXX در کل به معنای تکرار است تا موقعی که اتفاق خاصی رخ دهد ،مثلا تکرار یک کاری تا موقعی که ECX=0 شود ،دستور بعد از آن CMPS نام دارد، که به معنای مقایسه است ، Cmps دو عنصر رشته‌ای را تفریق کرده و فلگ‌ها را بر اساس آنها تنظیم می‌کند؛ (هیچکدام از عملوندها تغییر داده نمی‌شوند).

انواع فورمهای REPXX :

۱-repne :» تا زمانی که حالت مساوی وجود ندارد تکرار کن«

۲-repnz: »تا زمانی که صفر نیست تکرار کن«

این پیشوندها تا زمانی که CX صفر نیست، دستورالعمل اولیه را تکرار می‌کنند.

پیشوندهای repe و repz تکرار را تا زمانی که ZF=1 است تکرار می‌کنند.

پیشوندهای repne و repnz تکرار را تا زمانی که ZF=0 است تکرار می‌کنند.

پس معنی خط 4025BB اینطور است:

دو مقدار حافظه را به صورت بایت به بایت تا انتها با هم مقایسه کن

حالا به خط 4025B5 نگاه کنید ،بیایید در این خط یک BP بگذاریم و برنامه ری استارت کنیم و یک پسورد دلخواه وارد کنیم :

```
00402SA6  72 02          JB SHORT CrackmeC.004025AA
00402SA8  8BCD           MOV ECX,EBP
00402SAA  837C24 48 10   CMP DWORD PTR SS:[ESP+48],10
00402SAF  8B7424 34      MOV ESI,DWORD PTR SS:[ESP+34]
00402SB3  73 04          JNB SHORT CrackmeC.004025B9
00402SB5  8D7424 34      LEA ESI,DWORD PTR SS:[ESP+34]
00402SB9  33C0           XOR EAX,EAX                                  CrackmeC.0040E470
00402SBB  F3:A6          REPE CMPS BYTE PTR ES:[EDI],BYTE PTR DS:
00402SBD  74 05          JE SHORT CrackmeC.004025C4
00402SBF  1BC0           SBB EAX,EAX                                  CrackmeC.0040E470
00402SC1  83D8 FF        SBB EAX,-1
00402SC4  3BC3           CMP EAX,EBX
00402SC6  75 13          JNZ SHORT CrackmeC.004025DB
00402SC8  3BD5           CMP EDX,EBP
00402SCA  72 0F          JB SHORT CrackmeC.004025DB
00402SCC  33C0           XOR EAX,EAX                                  CrackmeC.0040E470
00402SCE  3BD5           CMP EDX,EBP
00402SD0  0F95C0         SETNE AL
00402SD3  3BC3           CMP EAX,EBX
00402SD5  0F84 CF020000  JE CrackmeC.004028AA
00402SDB  68 48E24000    PUSH CrackmeC.0040E248                       ASCII "I'm sorry but that is wrong!"
00402SE0  68 A02F4100    PUSH CrackmeC.00412FA0
00402SE5  E8 96FdFFFF    CALL CrackmeC.00401080

Stack address=0012FE88, (ASCII "12121212")
ESI=32313231
```

میبینید که Olly بروی خط مورد نظر متوقف شده :

LEA ESI, DWORD PTR SS:[ESP+34]

این خط آدرس موثر واقع در ثبات ESI را درون پشته قرار میدهد،که SS: به معنی پشته ، و [ESP+34] به معنای موقعیت پشته است ،اگر به فلش آبی رنگ در پنجره ی Info (بالای پنجره ی دامپ) نگاه کنید میبینید که SS:[ESP+34] با آدرس 12FE88 برابر است ،و محتوای این آدرس پسورد وارد شده ی ما است،حال یکبار کلید F8 را بفشارید و به مقدار ثبات ESI نگاه کنید (که اکنون بروی پشته است) :

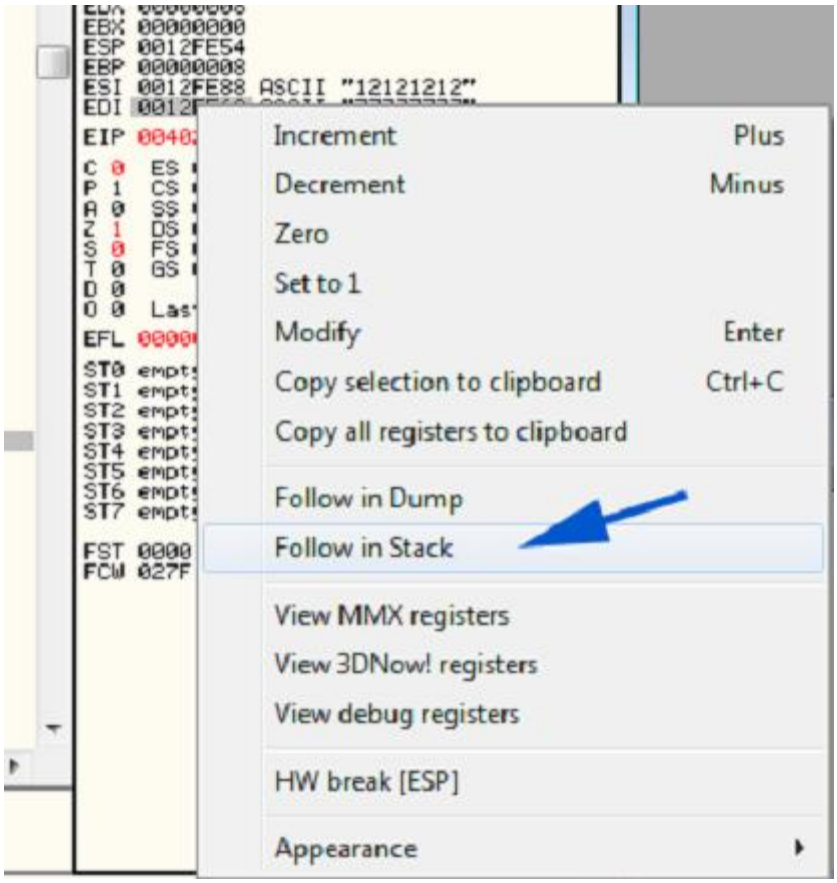
```
Registers (FPU)
EAX 0040E470 CrackmeC.0040E470
ECX 00000008
EDX 00000008
EBX 00000000
ESP 0012FE54
EBP 00000008
ESI 0012FE88 ASCII "12121212"
EDI 0012FE6C ASCII "77777777"
EIP 004025B9 CrackmeC.004025B9
C 1  ES 0023 32bit 0(FFFFFFFF)
P 1  CS 001B 32bit 0(FFFFFFFF)
```

ودر این آدرس حاوی پسورد ما است،دستور العمل بعدی مقدار EAXرا برابر 0 میکند ، و در خط بعدی ما دستور العمل REPE را داریم :

```
004025A8   8BCD          MOV ECX,EBP
004025AA   837C24 48 10  CMP DWORD PTR SS:[ESP+48],10
004025AF   8B7424 34     MOV ESI,DWORD PTR SS:[ESP+34]
004025B3 v 73 04         JNB SHORT CrackmeC.004025B9
004025B5   8D7424 34     LEA ESI,DWORD PTR SS:[ESP+34]
004025B9   33C0          XOR EAX,EAX
004025BB   F3:A6         REPE CMPS BYTE PTR ES:[EDI],BYTE PTR DS:
004025BD v 74 05         JE SHORT CrackmeC.004025C4
004025BF   1BC0          SBB EAX,EAX
004025C1   83D8 FF       SBB EAX,-1
004025C4   3BC3          CMP EAX,EBX
004025C6 v 75 13         JNZ SHORT CrackmeC.004025DB
004025C8   3BD5          CMP EDX,EBP
004025CA v 72 0F         JB SHORT CrackmeC.004025DB
004025CC   33C0          XOR EAX,EAX
004025CE   3BD5          CMP EDX,EBP
004025D0   0F95C0        SETNE AL
004025D3   3BC3          CMP EAX,EBX
004025D5 v 0F84 CF020000 JE CrackmeC.004028AA
004025DB   68 48E24000   PUSH CrackmeC.0040E248
004025E0   68 A02F4100   PUSH CrackmeC.00412FA0

ECX=00000008 (decimal 8.)
DS:[ESI]=stack [0012FE88]=31 ('1')
ES:[EDI]=stack [0012FE6C]=37 ('7')
```

همانطور که قبلا توضیح داده شد این دستور مقدار آدرس ذخیره شده در ESI را با مقدار آدرس حافظه ی EDI مقایسه میکند به صورت بایت به بایت (کاراکتر به کاراکتر) ،مقدار ثبات ECX یک واحد کاهش میابد ،و عملیات مقایسه از EDI و ESI ادامه میابد،و این حلقه تا موقعی که ECX=0 باشد ادامه میابد،اگر به تصویر زیر نگاه کنید میبینید که مقدار ECX برابر با 8 است (طول رمز وارد شده ی ما) و هر کارکتر از پسورد ما با مقدار EDI مقایسه میشود ،اما صبر کنید!! پسورد ما با چه چیزی مقایسه میشود؟اگر به ثبات EDI نگاه کنید آدرس پشته را میبینید که حاوی کد اسکی "7" است ،بروی ثبات EDI راست کلیک کنید و :"Follow in stack" :



پنجره ی پشته به آدرس مورد نظر رجوع میکند: 12FE6C و همانطور که مشاهده میکنید این آدرس حاوی رشته ی "37" است ،اگر به نمودار کد های اسکی نگاهی بیندازید میبینید که برابر با عدد "7" است :

```
0012FE6C   37373737
0012FE70   37373737
0012FE74   00412000  CrackmeC.00412000
0012FE78   0012FF88
0012FE7C   00000008
0012FE80   0000000F
0012FE84   00000000
0012FE88   32313231
0012FE8C   32313231
0012FE90   00406400  CrackmeC.00406400
0012FE94   003A0680
0012FE98   00000008
0012FE9C   0000000F
```

بله درست است رمز مورد نظر "77777777" است ،و دلیل مشکل طول پسورد هم در آدرس 4025C8 کاملا مشخص است به این صورت که ثبات EDX حاوی طول پسورد ما و ثبات EBP حاوی طول رمز واقعی است که در اولین سعی مان بسیار خوشانس بوده ایم که تعداد کاراکتر ها یکسان بوده ولی در سری بعدی که طول رمز کوتاه یا بلند تر بود پچ ما عمل نمیکرد ☺

حال بیایید برنامه را دوباره اجرا و رمز "77777777" را وارد کنید :

```
C:\Users\NODONGLE\Desktop\CrackmeConsole.exe
Welcome to AR Crackme #1
Please enter the correct serial:
7777777
```

وداریم:

```
C:\Users\NODONGLE\Desktop\CrackmeConsole.exe
Welcome to AR Crackme #1
Please enter the correct serial:
77777777
Checking serial
Congratulations that is correct!
Visit http://cracking.accessroot.com/ for everything AR
Long live the ARTeam!
Greetz & Shoutoutz to Whitefire for hosting our forum, forum.exetools.com, tech-
arena.com, and the ARTeam

Press any key to continue . . .
```

موفق شدیم ،اما هدف واقعی این آموزش بیان تفاوت بین روش LAME و NOOB بود ،پچ اولی بوسیله ی NOOBو کشف رمز روش LAME بود

کلام آخر:

اگر یادتان باشد ما توضیحات را یکی از روش هایی مدانیم که باعث میشود درک کلی از رفتار برنامه یا قسمتی از آن بهبود یابد،در زیر من همان قسمت کد را مستند سازی کرده ام ،ببینید چقدر راحت تر شده :

```
00402570  .  83C8 04        OR EAX,4
00402580  >  53             PUSH EBX
00402581  .  50             PUSH EAX
00402582  .  E8 100A0000    CALL CrackmeC.00402F97
00402587  >  837C24 2C 10   CMP DWORD PTR SS:[ESP+2C],10
0040258C  .  8B7C24 18      MOV EDI,DWORD PTR SS:[ESP+18]      ### EDI = HC password
00402590  .v 73 04          JNB SHORT CrackmeC.00402596        ### X
00402592  .  8D7C24 18      LEA EDI,DWORD PTR SS:[ESP+18]      ### EDI = HC password
00402596  >  8B5424 44      MOV EDX,DWORD PTR SS:[ESP+44]      ### EDX = Password length
0040259A  .  3BD3           CMP EDX,EBX                        ### CMP length with zero
0040259C  .  8B6C24 28      MOV EBP,DWORD PTR SS:[ESP+28]      ### EBP = length of HC password
004025A0  .v 74 26          JE SHORT CrackmeC.004025C8         ### Jump if password zero length
004025A2  .  3BD5           CMP EDX,EBP                        ### Is length < hard coded amount (8 - [esp+28])
004025A4  .  8BCA           MOV ECX,EDX                        ### ECX = length
004025A6  .v 72 02          JB SHORT CrackmeC.004025AA         ### X Jmp if our length is < hard coded length (8)
004025A8  .  8BCD           MOV ECX,EBP                        ### ECX = Length of HC password
004025AA  >  837C24 48 10   CMP DWORD PTR SS:[ESP+48],10       ### CMP 0x0F with 0x10 ???
004025AF  .  8B7424 34      MOV ESI,DWORD PTR SS:[ESP+34]      ### MOV First digits of entered password into ESI
004025B3  .v 73 04          JNB SHORT CrackmeC.004025B9        ### X
004025B5  .  8D7424 34      LEA ESI,DWORD PTR SS:[ESP+34]      ### ESI = entered password
004025B9  >  33C0           XOR EAX,EAX                        ### EAX = 0
004025BB  .  F3:A6          REPE CMPS BYTE PTR ES:[EDI],BYTE PTR DS  ### CMP H.C. password with entered passsword
004025BD  .v 74 05          JE SHORT CrackmeC.004025C4         ### JMP if they are the same
004025BF  .  1BC0           SBB EAX,EAX                        ### EAX = FFFFFFFF
004025C1  .  83D8 FF        SBB EAX,-1                         ### No change to EAX
004025C4  >  3BC3           CMP EAX,EBX                        ### EAX (FFFFFFFF) == EBX (0)
004025C6  .v 75 19          JNZ SHORT CrackmeC.004025DB        ### <> !!!!!
004025C8  >  3BD5           CMP EDX,EBP
004025CA  .v 72 0F          JB SHORT CrackmeC.004025DB         ### <> !!!!!
004025CC  .  33C0           XOR EAX,EAX
004025CE  .  3BD5           CMP EDX,EBP
004025D0  .  0F95C0         SETNE AL
004025D3  .  3BC3           CMP EAX,EBX
004025D5  .v 0F84 CF020000  JE CrackmeC.004028AA
004025DB  >  68 48E24000    PUSH CrackmeC.0040E248             ASCII "I'm sorry but that is wrong!"
004025E0  .  68 A02F4100    PUSH CrackmeC.00412FA0
004025E5  .  E8 A6F4FFFF    CALL CrackmeC.00401A90
004025EA  .  83C4 08        ADD ESP,8
004025ED  .  8BF0           MOV ESI,EAX
004025EF  .  6A 0A          PUSH 0A
```