Analyzing CrackMe2 with String Decryption

Objective: Learn to analyze a CrackMe challenge with obfuscated strings by using breakpoints to observe runtime decryption, then identify and bypass the password validation logic.

Introduction: What Makes This CrackMe Different?

Unlike CrackMe1, which had plaintext strings, **CrackMe2 uses string obfuscation**. This means:

- Strings are encrypted in the executable
- They are decrypted at runtime (when the program runs)
- We cannot find password messages by simply searching for strings
- We must observe the program's behavior dynamically

Why do programs obfuscate strings?

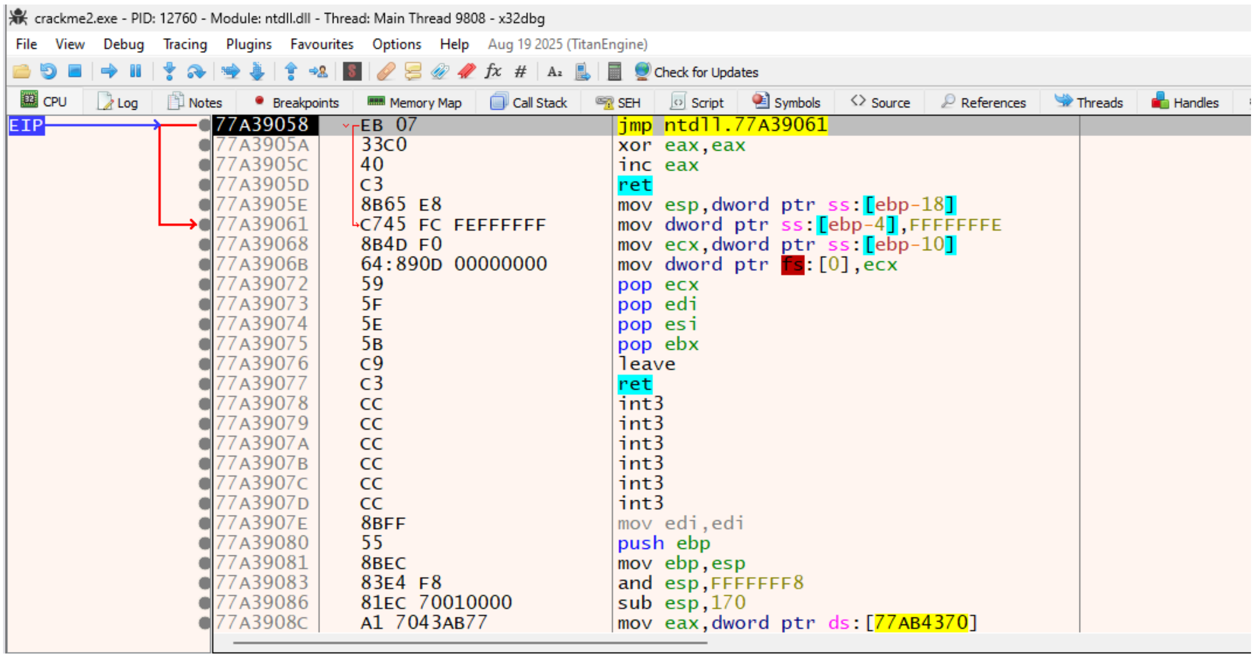
- Hide functionality from static analysis
- Prevent detection by antivirus
- Protect intellectual property
- Make reverse engineering more difficult

Step 1: Load the Executable into the Debugger

Instructions:

1. **Open your debugger** (x32dbg or OllyDbg)
2. **Load CrackMe2.exe:**
   - File → Open
   - Navigate to the CrackMe2.exe file
   - Click Open
3. **Observe the initial state:**
   - The debugger pauses at the system initialization code
   - Notice the title bar displays **"Module: ntdll.dll"**

What you're seeing: The Windows system loader (ntdll.dll) preparing the program to run. This is **not** the actual program code yet.



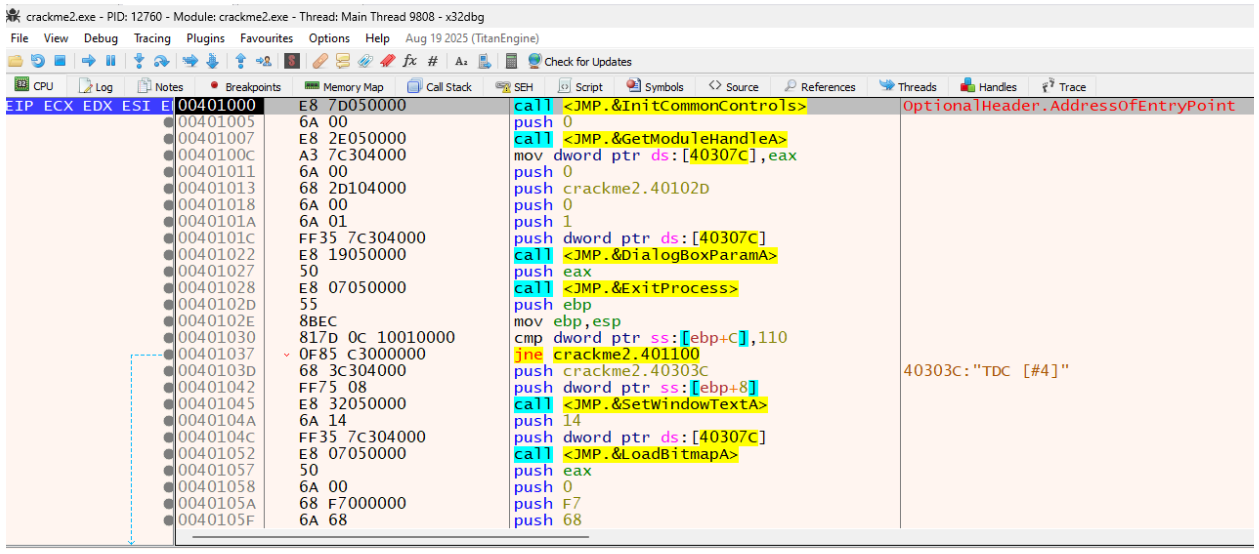
```
77A39058  EB 07              jmp ntdll.77A39061
77A3905A  33C0               xor eax,eax
77A3905C  40                 inc eax
77A3905D  C3                 ret
77A3905E  8B65 E8            mov esp,dword ptr ss:[ebp-18]
77A39061  C745 FC FEFFFFFF   mov dword ptr ss:[ebp-4],FFFFFFFE
77A39068  8B4D F0            mov ecx,dword ptr ss:[ebp-10]
77A3906B  64:890D 00000000   mov dword ptr fs:[0],ecx
77A39072  59                 pop ecx
77A39073  5F                 pop edi
77A39074  5E                 pop esi
77A39075  5B                 pop ebx
77A39076  C9                 leave
77A39077  C3                 ret
77A39078  CC                 int3
77A39079  CC                 int3
77A3907A  CC                 int3
77A3907B  CC                 int3
77A3907C  CC                 int3
77A3907D  CC                 int3
77A3907E  8BFF               mov edi,edi
77A39080  55                 push ebp
77A39081  8BEC               mov ebp,esp
77A39083  83E4 F8            and esp,FFFFFFF8
77A39086  81EC 70010000      sub esp,170
77A3908C  A1 7043AB77        mov eax,dword ptr ds:[77AB4370]
```

Step 2: Navigate to the Program Entry Point

Instructions:

1. **Run to the Address of Entry Point (AEP):**
   - Press **F9** (or click the blue **Run** button ▶)
   - The debugger will execute through system initialization
   - Execution will pause at the program's first instruction
2. **Verify you're at the correct location:**
   - Check the title bar: It should now show **"Module: CrackMe2.exe"** (or similar)
   - The CPU view should display the program's actual code

Important: The "Module:" section changing from "ntdll.dll" to "CrackMe2.exe" confirms you're now analyzing the actual program, not Windows system code.


File View Debug Tracing Plugins Favourites Options Help Aug 19 2025 (TitanEngine)
Check for Updates
CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads Handles Trace

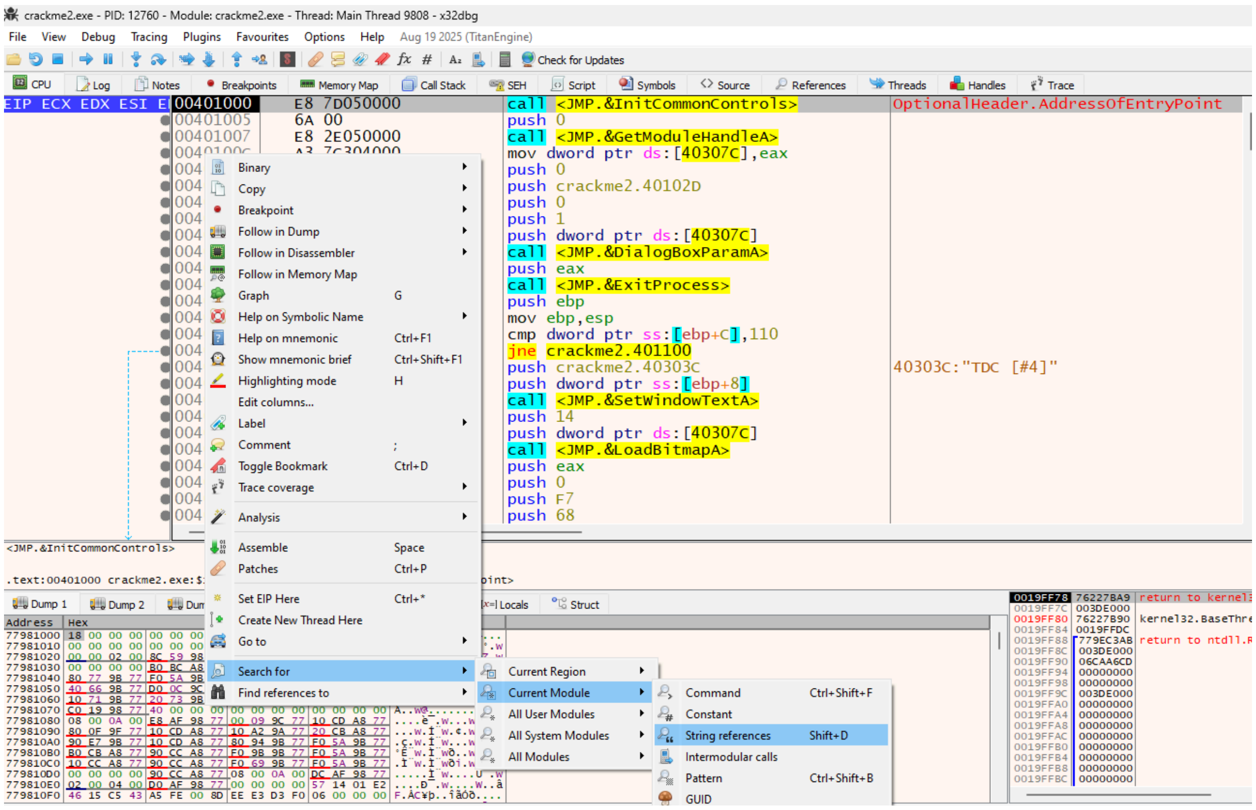
crackme2.exe - PID: 12760 - Module: crackme2.exe - Thread: Main Thread 9808 - x32dbg

```
EIP ECX EDX ESI E 00401000  E8 7D050000        call <JMP.&InitCommonControls>     OptionalHeader.AddressOfEntryPoint
                  00401005  6A 00              push 0
                  00401007  E8 2E050000        call <JMP.&GetModuleHandleA>
                  0040100C  A3 7C304000        mov dword ptr ds:[40307C],eax
                  00401011  6A 00              push 0
                  00401013  68 2D104000        push crackme2.40102D
                  00401018  6A 00              push 0
                  0040101A  6A 01              push 1
                  0040101C  FF35 7C304000      push dword ptr ds:[40307C]
                  00401022  E8 19050000        call <JMP.&DialogBoxParamA>
                  00401027  50                 push eax
                  00401028  E8 07050000        call <JMP.&ExitProcess>
                  0040102D  55                 push ebp
                  0040102E  8BEC               mov ebp,esp
                  00401030  817D 0C 10010000   cmp dword ptr ss:[ebp+C],110
                  00401037  0F85 C3000000      jne crackme2.401100
                  0040103D  68 3C304000        push crackme2.40303C               40303C:"TDC [#4]"
                  00401042  FF75 08            push dword ptr ss:[ebp+8]
                  00401045  E8 32050000        call <JMP.&SetWindowTextA>
                  0040104A  6A 14              push 14
                  0040104C  FF35 7C304000      push dword ptr ds:[40307C]
                  00401052  E8 07050000        call <JMP.&LoadBitmapA>
                  00401057  50                 push eax
                  00401058  6A 00              push 0
                  0040105A  68 F7000000        push F7
                  0040105F  6A 68              push 68
```

Step 3: Search for Strings

Instructions:

1. **Open the strings window:**
    - While at the Address of Entry Point, press **Shift+D**
    - *Alternative: Right-click → Search for → String references*
    - A window will appear showing all strings found in the executable



```
EIP ECX EDX ESI E 00401000   E8 7D050000   call <JMP.&InitCommonControls>    OptionalHeader.AddressOfEntryPoint
                  00401005   6A 00         push 0
                  00401007   E8 2E050000   call <JMP.&GetModuleHandleA>
                                           mov dword ptr ds:[40307C],eax
                                           push 0
                                           push crackme2.40102D
                                           push 0
                                           push 1
                                           push dword ptr ds:[40307C]
                                           call <JMP.&DialogBoxParamA>
                                           push eax
                                           call <JMP.&ExitProcess>
                                           push ebp
                                           mov ebp,esp
                                           cmp dword ptr ss:[ebp+C],110
                                           jne crackme2.401100
                                           push crackme2.40303C              40303C:"TDC [#4]"
                                           push dword ptr ss:[ebp+8]
                                           call <JMP.&SetWindowTextA>
                                           push 14
                                           push dword ptr ds:[40307C]
                                           call <JMP.&LoadBitmapA>
                                           push eax
                                           push 0
                                           push F7
                                           push 68

<JMP.&InitCommonControls>

.text:00401000 crackme2.exe:$
```

```
0019FF78  76227BA9  return to kernel3
0019FF7C  003DE000
0019FF80  76227B90  kernel32.BaseThre
0019FF84  0019FFDC
0019FF88  779EC3AB  return to ntdll.R
0019FF8C  003DE000
0019FF90  06CAA6CD
0019FF94  00000000
0019FF98  00000000
0019FF9C  003DE000
0019FFA0  00000000
0019FFA4  00000000
0019FFA8  00000000
0019FFAC  00000000
0019FFB0  00000000
0019FFB4  00000000
0019FFB8  00000000
0019FFBC  00000000
```

2. **Observe the strings:**
    - You'll notice many strings appear **obfuscated or encrypted**
    - These are **NOT** readable text - they're encrypted!
3. **What you WON'T find:**
    - Clear messages like "Correct password!" or "Access denied!"
    - Obvious validation logic
    - Readable error messages

Why strings are obfuscated: The program encrypts strings at compile time and decrypts them at runtime. This prevents analysts from finding interesting code by searching for obvious strings.

Since we cannot identify interesting strings by reading them, we need a different approach: **observe the program decrypt them at runtime**.

Step 4: Strategic Breakpoint Placement

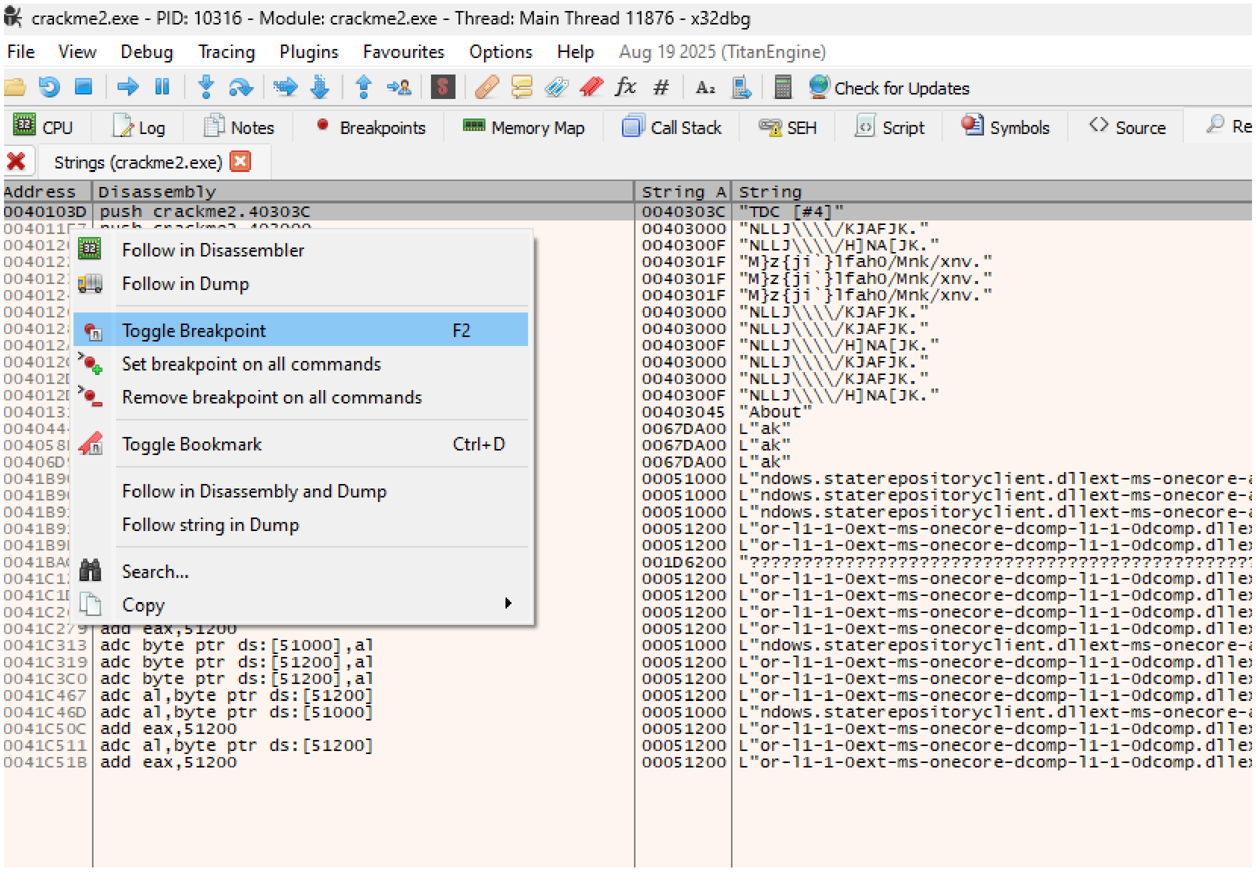
The Strategy:

Since we don't know which strings are important, we'll:

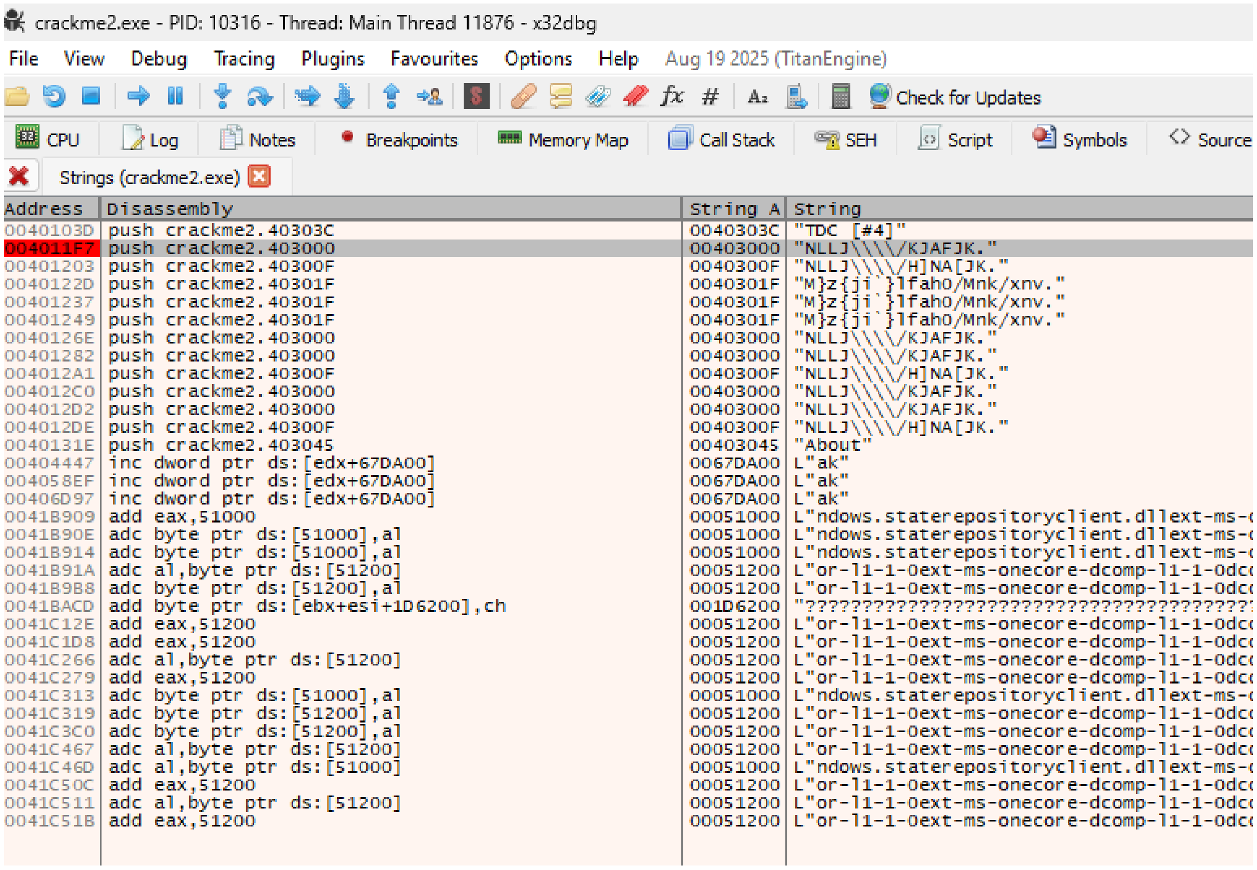
1. Set breakpoints on **unique encrypted strings**
2. Run the program and let it hit each breakpoint
3. Observe if the string gets decrypted
4. Identify what the decrypted string reveals

Instructions:

1. **In the strings window (Shift+D):**
    - Identify unique encrypted strings (avoid duplicates)
    - Select diverse strings from different parts of the program
2. **Set breakpoints on strings: Method A - From the strings window:**
    - **Double-click** on an encrypted string
    - This takes you to where that string is referenced in the code
    - Press **F2** to set a breakpoint at that location
    - The line will be highlighted (usually red)
3. **Method B - Right-click method:**
    - Right-click on the string reference
    - Select "Toggle Breakpoint" or "Set Breakpoint"



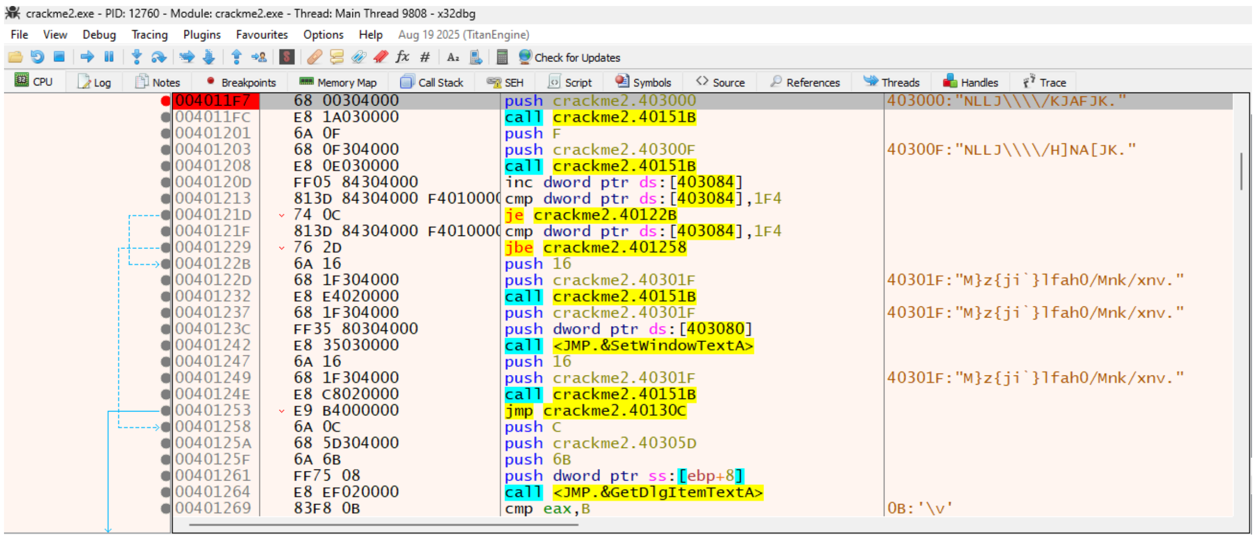
```
Address  Disassembly                       String A String
0040103D push crackme2.40303C              0040303C "TDC [#4]"
004011F7 push crackme2.403000              00403000 "NLLJ\\\\/KJAFJK."
004012                                     0040300F "NLLJ\\\\/H]NA[JK."
004012                                     0040301F "M}z{ji`}lfahO/Mnk/xnv."
004012                                     0040301F "M}z{ji`}lfahO/Mnk/xnv."
004012                                     0040301F "M}z{ji`}lfahO/Mnk/xnv."
004012                                     00403000 "NLLJ\\\\/KJAFJK."
004012                                     00403000 "NLLJ\\\\/KJAFJK."
004012                                     0040300F "NLLJ\\\\/H]NA[JK."
004012                                     00403000 "NLLJ\\\\/KJAFJK."
004012D                                    00403000 "NLLJ\\\\/KJAFJK."
004012D                                    0040300F "NLLJ\\\\/H]NA[JK."
004013                                     00403045 "About"
004044                                     0067DA00 L"ak"
004058                                     0067DA00 L"ak"
00406D                                     0067DA00 L"ak"
0041B9                                     00051000 L"ndows.staterepositoryclient.dllext-ms-onecore-
0041B9                                     00051000 L"ndows.staterepositoryclient.dllext-ms-onecore-
0041B9                                     00051000 L"ndows.staterepositoryclient.dllext-ms-onecore-
0041B9                                     00051200 L"or-l1-1-0ext-ms-onecore-dcomp-l1-1-0dcomp.dlle
0041B9                                     00051200 L"or-l1-1-0ext-ms-onecore-dcomp-l1-1-0dcomp.dlle
0041BA                                     001D6200 "?????????????????????????????????????????????
0041C1                                     00051200 L"or-l1-1-0ext-ms-onecore-dcomp-l1-1-0dcomp.dlle
0041C1                                     00051200 L"or-l1-1-0ext-ms-onecore-dcomp-l1-1-0dcomp.dlle
0041C2                                     00051200 L"or-l1-1-0ext-ms-onecore-dcomp-l1-1-0dcomp.dlle
0041C279 add eax,51200                     00051200 L"or-l1-1-0ext-ms-onecore-dcomp-l1-1-0dcomp.dlle
0041C313 adc byte ptr ds:[51000],al        00051000 L"ndows.staterepositoryclient.dllext-ms-onecore-
0041C319 adc byte ptr ds:[51200],al        00051200 L"or-l1-1-0ext-ms-onecore-dcomp-l1-1-0dcomp.dlle
0041C3C0 adc byte ptr ds:[51200],al        00051200 L"or-l1-1-0ext-ms-onecore-dcomp-l1-1-0dcomp.dlle
0041C467 adc al,byte ptr ds:[51200]        00051200 L"or-l1-1-0ext-ms-onecore-dcomp-l1-1-0dcomp.dlle
0041C46D adc al,byte ptr ds:[51000]        00051000 L"ndows.staterepositoryclient.dllext-ms-onecore-
0041C50C add eax,51200                     00051200 L"or-l1-1-0ext-ms-onecore-dcomp-l1-1-0dcomp.dlle
0041C511 adc al,byte ptr ds:[51200]        00051200 L"or-l1-1-0ext-ms-onecore-dcomp-l1-1-0dcomp.dlle
0041C51B add eax,51200                     00051200 L"or-l1-1-0ext-ms-onecore-dcomp-l1-1-0dcomp.dlle
```



crackme2.exe - PID: 10316 - Thread: Main Thread 11876 - x32dbg

File View Debug Tracing Plugins Favourites Options Help Aug 19 2025 (TitanEngine)
Check for Updates
CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source
Strings (crackme2.exe)

```
Address   Disassembly                                String A String
0040103D  push crackme2.40303C                        0040303C "TDC [#4]"
004011F7  push crackme2.403000                        00403000 "NLLJ\\\\/KJAFJK."
00401203  push crackme2.40300F                        0040300F "NLLJ\\\\/H]NA[JK."
0040122D  push crackme2.40301F                        0040301F "M}z{ji`}lfah0/Mnk/xnv."
00401237  push crackme2.40301F                        0040301F "M}z{ji`}lfah0/Mnk/xnv."
00401249  push crackme2.40301F                        0040301F "M}z{ji`}lfah0/Mnk/xnv."
0040126E  push crackme2.403000                        00403000 "NLLJ\\\\/KJAFJK."
00401282  push crackme2.403000                        00403000 "NLLJ\\\\/KJAFJK."
004012A1  push crackme2.40300F                        0040300F "NLLJ\\\\/H]NA[JK."
004012C0  push crackme2.403000                        00403000 "NLLJ\\\\/KJAFJK."
004012D2  push crackme2.403000                        00403000 "NLLJ\\\\/KJAFJK."
004012DE  push crackme2.40300F                        0040300F "NLLJ\\\\/H]NA[JK."
0040131E  push crackme2.403045                        00403045 "About"
00404447  inc dword ptr ds:[edx+67DA00]               0067DA00 L"ak"
004058EF  inc dword ptr ds:[edx+67DA00]               0067DA00 L"ak"
00406D97  inc dword ptr ds:[edx+67DA00]               0067DA00 L"ak"
0041B909  add eax,51000                               00051000 L"ndows.staterepositoryclient.dllext-ms-
0041B90E  adc byte ptr ds:[51000],al                  00051000 L"ndows.staterepositoryclient.dllext-ms-
0041B914  adc byte ptr ds:[51000],al                  00051000 L"ndows.staterepositoryclient.dllext-ms-
0041B91A  adc al,byte ptr ds:[51200]                  00051200 L"or-l1-1-0ext-ms-onecore-dcomp-l1-1-0dc
0041B9B8  adc byte ptr ds:[51200],al                  00051200 L"or-l1-1-0ext-ms-onecore-dcomp-l1-1-0dc
0041BACD  add byte ptr ds:[ebx+esi+1D6200],ch         001D6200 "?????????????????????????????????????
0041C12E  add eax,51200                               00051200 L"or-l1-1-0ext-ms-onecore-dcomp-l1-1-0dc
0041C1D8  add eax,51200                               00051200 L"or-l1-1-0ext-ms-onecore-dcomp-l1-1-0dc
0041C266  adc al,byte ptr ds:[51200]                  00051200 L"or-l1-1-0ext-ms-onecore-dcomp-l1-1-0dc
0041C279  add eax,51200                               00051200 L"or-l1-1-0ext-ms-onecore-dcomp-l1-1-0dc
0041C313  adc byte ptr ds:[51000],al                  00051000 L"ndows.staterepositoryclient.dllext-ms-
0041C319  adc byte ptr ds:[51200],al                  00051200 L"or-l1-1-0ext-ms-onecore-dcomp-l1-1-0dc
0041C3C0  adc byte ptr ds:[51200],al                  00051200 L"or-l1-1-0ext-ms-onecore-dcomp-l1-1-0dc
0041C467  adc al,byte ptr ds:[51200]                  00051200 L"or-l1-1-0ext-ms-onecore-dcomp-l1-1-0dc
0041C46D  adc al,byte ptr ds:[51000]                  00051000 L"ndows.staterepositoryclient.dllext-ms-
0041C50C  add eax,51200                               00051200 L"or-l1-1-0ext-ms-onecore-dcomp-l1-1-0dc
0041C511  adc al,byte ptr ds:[51200]                  00051200 L"or-l1-1-0ext-ms-onecore-dcomp-l1-1-0dc
0041C51B  add eax,51200                               00051200 L"or-l1-1-0ext-ms-onecore-dcomp-l1-1-0dc
```



crackme2.exe - PID: 12760 - Module: crackme2.exe - Thread: Main Thread 9808 - x32dbg

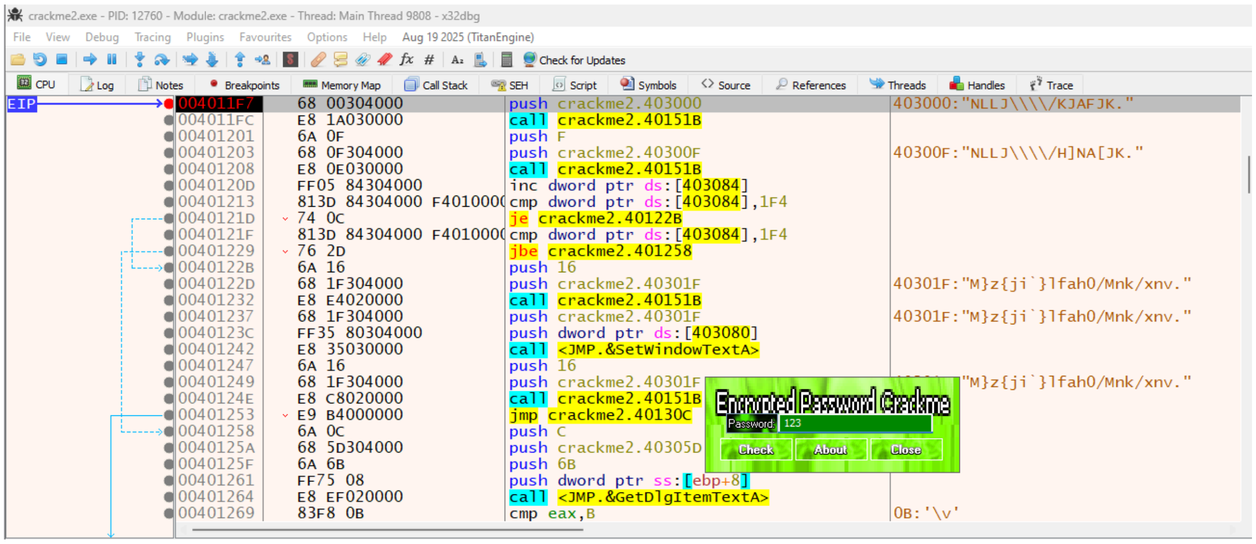
File View Debug Tracing Plugins Favourites Options Help Aug 19 2025 (TitanEngine)
Check for Updates
CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads Handles Trace

```
004011F7    68 00304000          push crackme2.403000               403000:"NLLJ\\\\/KJAFJK."
004011FC    E8 1A030000          call crackme2.40151B
00401201    6A 0F                push F
00401203    68 0F304000          push crackme2.40300F               40300F:"NLLJ\\\\/H]NA[JK."
00401208    E8 0E030000          call crackme2.40151B
0040120D    FF05 84304000        inc dword ptr ds:[403084]
00401213    813D 84304000 F401000( cmp dword ptr ds:[403084],1F4
0040121D  ˅ 74 0C                je crackme2.40122B
0040121F    813D 84304000 F401000( cmp dword ptr ds:[403084],1F4
00401229  ˅ 76 2D                jbe crackme2.401258
0040122B    6A 16                push 16
0040122D    68 1F304000          push crackme2.40301F               40301F:"M}z{ji`}lfah0/Mnk/xnv."
00401232    E8 E4020000          call crackme2.40151B
00401237    68 1F304000          push crackme2.40301F               40301F:"M}z{ji`}lfah0/Mnk/xnv."
0040123C    FF35 80304000        push dword ptr ds:[403080]
00401242    E8 35030000          call <JMP.&SetWindowTextA>
00401247    6A 16                push 16
00401249    68 1F304000          push crackme2.40301F               40301F:"M}z{ji`}lfah0/Mnk/xnv."
0040124E    E8 C8020000          call crackme2.40151B
00401253  ˅ E9 B4000000          jmp crackme2.40130C
00401258    6A 0C                push C
0040125A    68 5D304000          push crackme2.40305D
0040125F    6A 6B                push 6B
00401261    FF75 08              push dword ptr ss:[ebp+8]
00401264    E8 EF020000          call <JMP.&GetDlgItemTextA>
00401269    83F8 0B              cmp eax,B                          0B:'\v'
```

Step 5: Execute to First Breakpoint

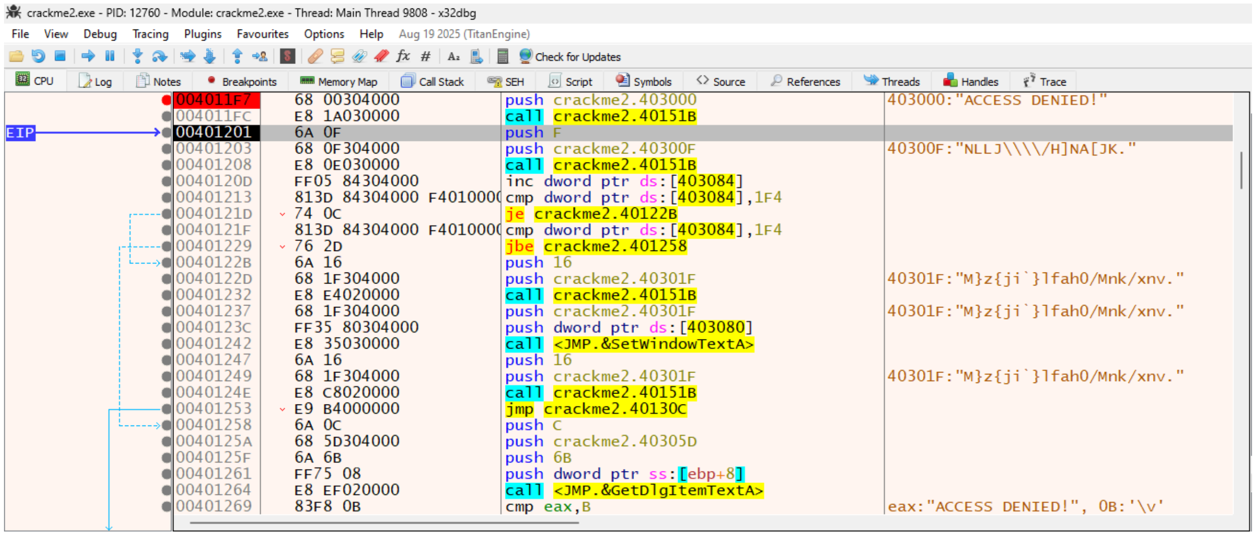
Instructions:

1. **Run the program:**
   - Press **F9** (Run)
   - The program will execute until it hits your first breakpoint
2. **The program window may appear:**
   - The CrackMe2 GUI might display
   - It may prompt for a password
   - I have entered 123 as a password


crackme2.exe - PID: 12760 - Module: crackme2.exe - Thread: Main Thread 9808 - x32dbg
File View Debug Tracing Plugins Favourites Options Help Aug 19 2025 (TitanEngine)
Check for Updates
CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads Handles Trace
EIP
Encrypted Password Crackme
Password: 123
Check
About
Close

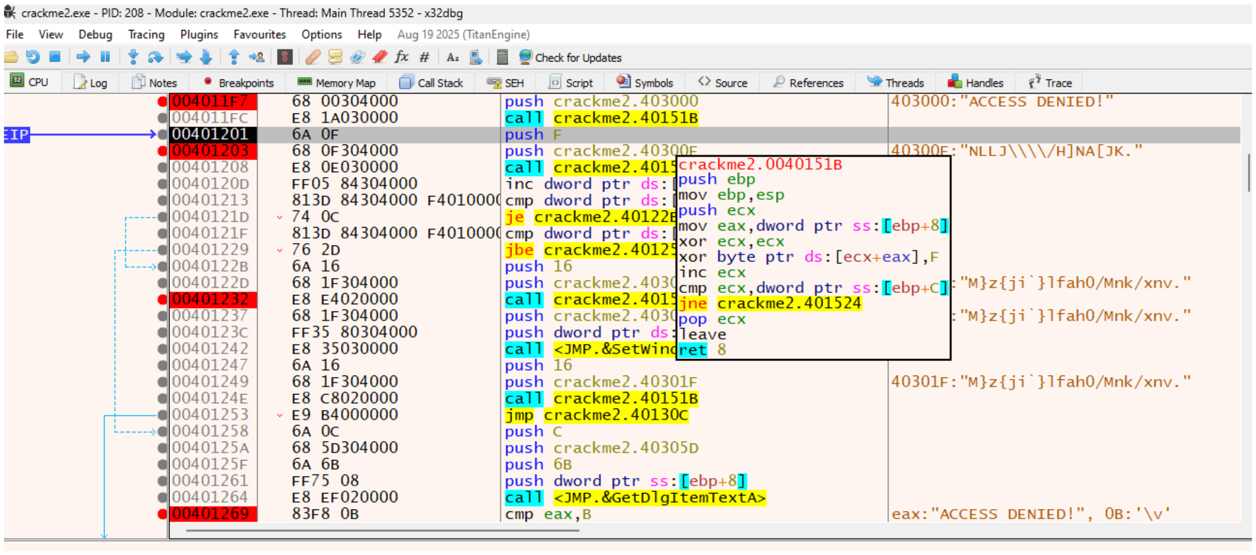
```
004011F7  68 00304000          push crackme2.403000            403000:"NLLJ\\\\/KJAFJK."
004011FC  E8 1A030000          call crackme2.40151B
00401201  6A 0F                push F
00401203  68 0F304000          push crackme2.40300F            40300F:"NLLJ\\\\/H]NA[JK."
00401208  E8 0E030000          call crackme2.40151B
0040120D  FF05 84304000        inc dword ptr ds:[403084]
00401213  813D 84304000 F4010000 cmp dword ptr ds:[403084],1F4
0040121D  74 0C                je crackme2.40122B
0040121F  813D 84304000 F4010000 cmp dword ptr ds:[403084],1F4
00401229  76 2D                jbe crackme2.401258
0040122B  6A 16                push 16
0040122D  68 1F304000          push crackme2.40301F            40301F:"M}z{ji`}lfah0/Mnk/xnv."
00401232  E8 E4020000          call crackme2.40151B
00401237  68 1F304000          push crackme2.40301F            40301F:"M}z{ji`}lfah0/Mnk/xnv."
0040123C  FF35 80304000        push dword ptr ds:[403080]
00401242  E8 35030000          call <JMP.&SetWindowTextA>
00401247  6A 16                push 16
00401249  68 1F304000          push crackme2.40301F            "M}z{ji`}lfah0/Mnk/xnv."
0040124E  E8 C8020000          call crackme2.40151B
00401253  E9 B4000000          jmp crackme2.40130C
00401258  6A 0C                push C
0040125A  68 5D304000          push crackme2.40305D
0040125F  6A 6B                push 6B
00401261  FF75 08              push dword ptr ss:[ebp+8]
00401264  E8 EF020000          call <JMP.&GetDlgItemTextA>
00401269  83F8 0B              cmp eax,B                       0B:'\v'
```

3. **Execution pauses at first breakpoint:**
   - The debugger stops at the first string reference you marked
   - The highlighted line shows where you are in the code
4. **Observe the encrypted string:**
   - Look at the string at this location
   - Note that it's still **encrypted/obfuscated**
   - Example: `"N@@SSS\/H]NA[JK."`


crackme2.exe - PID: 12760 - Module: crackme2.exe - Thread: Main Thread 9808 - x32dbg
File View Debug Tracing Plugins Favourites Options Help Aug 19 2025 (TitanEngine)
Check for Updates
CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads Handles Trace
EIP

```
004011F7  68 00304000          push crackme2.403000             403000:"ACCESS DENIED!"
004011FC  E8 1A030000          call crackme2.40151B
00401201  6A 0F                push F
00401203  68 0F304000          push crackme2.40300F             40300F:"NLLJ\\\\/H]NA[JK."
00401208  E8 0E030000          call crackme2.40151B
0040120D  FF05 84304000        inc dword ptr ds:[403084]
00401213  813D 84304000 F401000( cmp dword ptr ds:[403084],1F4
0040121D  74 0C                je crackme2.40122B
0040121F  813D 84304000 F401000( cmp dword ptr ds:[403084],1F4
00401229  76 2D                jbe crackme2.401258
0040122B  6A 16                push 16
0040122D  68 1F304000          push crackme2.40301F             40301F:"M}z{ji`}lfah0/Mnk/xnv."
00401232  E8 E4020000          call crackme2.40151B
00401237  68 1F304000          push crackme2.40301F             40301F:"M}z{ji`}lfah0/Mnk/xnv."
0040123C  FF35 80304000        push dword ptr ds:[403080]
00401242  E8 35030000          call <JMP.&SetWindowTextA>
00401247  6A 16                push 16
00401249  68 1F304000          push crackme2.40301F             40301F:"M}z{ji`}lfah0/Mnk/xnv."
0040124E  E8 C8020000          call crackme2.40151B
00401253  E9 B4000000          jmp crackme2.40130C
00401258  6A 0C                push C
0040125A  68 5D304000          push crackme2.40305D
0040125F  6A 6B                push 6B
00401261  FF75 08              push dword ptr ss:[ebp+8]
00401264  E8 EF020000          call <JMP.&GetDlgItemTextA>
00401269  83F8 0B              cmp eax,B                        eax:"ACCESS DENIED!", 0B:'\v'
```

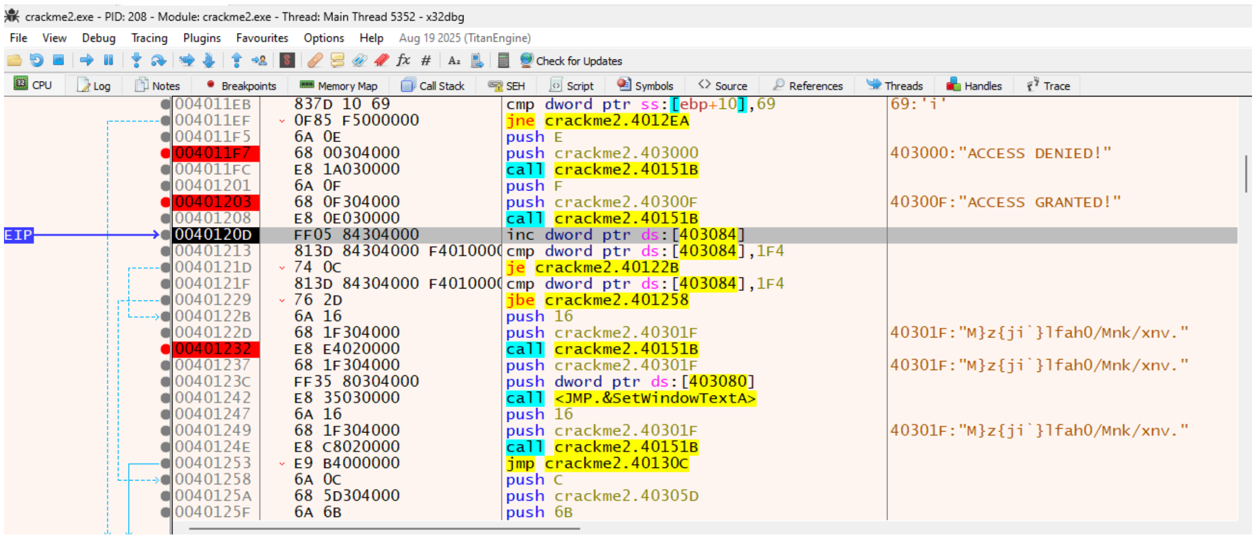
It seems the function between our previous string breakpoint and our current one, crackme2.40151B, decrypted the string


crackme2.exe - PID: 208 - Module: crackme2.exe - Thread: Main Thread 5352 - x32dbg
File View Debug Tracing Plugins Favourites Options Help Aug 19 2025 (TitanEngine)
Check for Updates
CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads Handles Trace
EIP

```
004011F7  68 00304000          push crackme2.403000             403000:"ACCESS DENIED!"
004011FC  E8 1A030000          call crackme2.40151B
00401201  6A 0F                push F
00401203  68 0F304000          push crackme2.40300F             40300F:"NLLJ\\\\/H]NA[JK."
00401208  E8 0E030000          call crackme2.401
0040120D  FF05 84304000        inc dword ptr ds:
00401213  813D 84304000 F401000( cmp dword ptr ds:
0040121D  74 0C                je crackme2.40122B
0040121F  813D 84304000 F401000( cmp dword ptr ds:
00401229  76 2D                jbe crackme2.4012
0040122B  6A 16                push 16
0040122D  68 1F304000          push crackme2.4030               :"M}z{ji`}lfah0/Mnk/xnv."
00401232  E8 E4020000          call crackme2.401
00401237  68 1F304000          push crackme2.4030               :"M}z{ji`}lfah0/Mnk/xnv."
0040123C  FF35 80304000        push dword ptr ds
00401242  E8 35030000          call <JMP.&SetWind
00401247  6A 16                push 16
00401249  68 1F304000          push crackme2.40301F             40301F:"M}z{ji`}lfah0/Mnk/xnv."
0040124E  E8 C8020000          call crackme2.40151B
00401253  E9 B4000000          jmp crackme2.40130C
00401258  6A 0C                push C
0040125A  68 5D304000          push crackme2.40305D
0040125F  6A 6B                push 6B
00401261  FF75 08              push dword ptr ss:[ebp+8]
00401264  E8 EF020000          call <JMP.&GetDlgItemTextA>
00401269  83F8 0B              cmp eax,B                        eax:"ACCESS DENIED!", 0B:'\v'
```

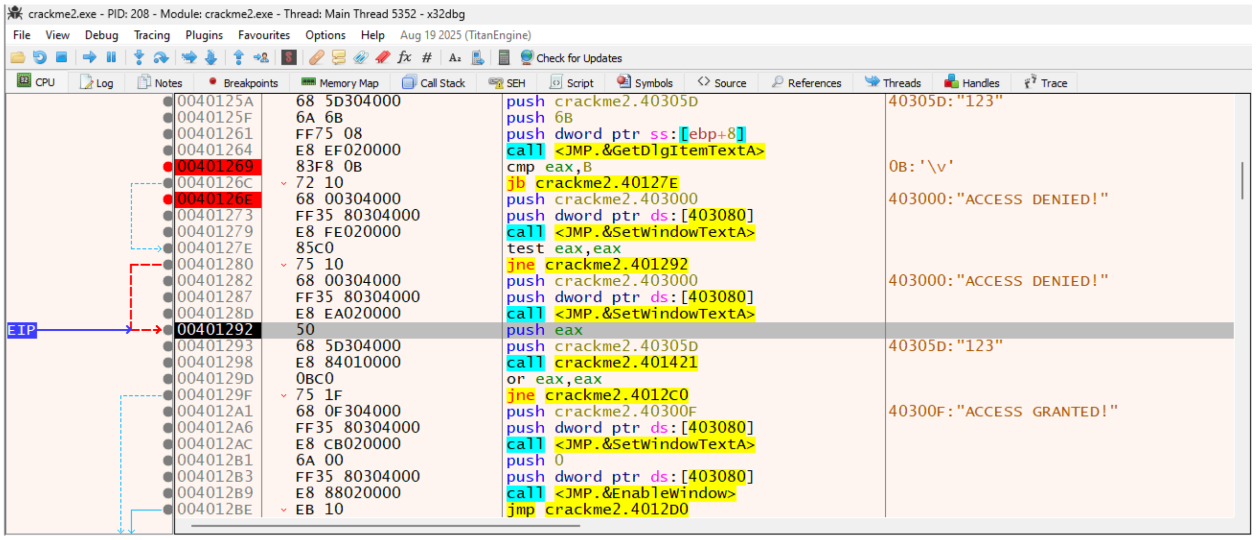
```
crackme2.0040151B
push ebp
mov ebp,esp
push ecx
mov eax,dword ptr ss:[ebp+8]
xor ecx,ecx
xor byte ptr ds:[ecx+eax],F
inc ecx
cmp ecx,dword ptr ss:[ebp+C]
jne crackme2.401524
pop ecx
leave
ret 8
```

From this, we can see that the encrypted text, “NLLJ\\\\/H]NA[JK.” was actually “ACCESS GRANTED!” Perfect.


crackme2.exe - PID: 208 - Module: crackme2.exe - Thread: Main Thread 5352 - x32dbg
File View Debug Tracing Plugins Favourites Options Help Aug 19 2025 (TitanEngine)
Check for Updates
CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads Handles Trace
EIP

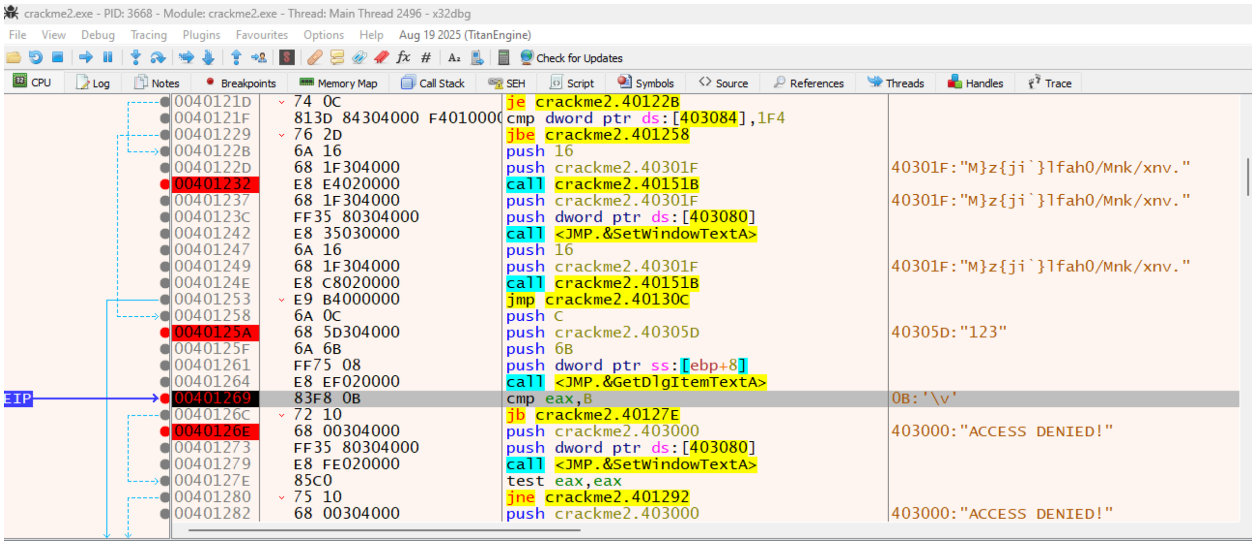
```
004011EB    837D 10 69          cmp dword ptr ss:[ebp+10],69      69:'i'
004011EF    0F85 F5000000       jne crackme2.4012EA
004011F5    6A 0E               push E
004011F7    68 00304000         push crackme2.403000              403000:"ACCESS DENIED!"
004011FC    E8 1A030000         call crackme2.40151B
00401201    6A 0F               push F
00401203    68 0F304000         push crackme2.40300F              40300F:"ACCESS GRANTED!"
00401208    E8 0E030000         call crackme2.40151B
0040120D    FF05 84304000       inc dword ptr ds:[403084]
00401213    813D 84304000 F4010000 cmp dword ptr ds:[403084],1F4
0040121D    74 0C               je crackme2.40122B
0040121F    813D 84304000 F4010000 cmp dword ptr ds:[403084],1F4
00401229    76 2D               jbe crackme2.401258
0040122B    6A 16               push 16
0040122D    68 1F304000         push crackme2.40301F              40301F:"M}z{ji`}lfah0/Mnk/xnv."
00401232    E8 E4020000         call crackme2.40151B
00401237    68 1F304000         push crackme2.40301F              40301F:"M}z{ji`}lfah0/Mnk/xnv."
0040123C    FF35 80304000       push dword ptr ds:[403080]
00401242    E8 35030000         call <JMP.&SetWindowTextA>
00401247    6A 16               push 16
00401249    68 1F304000         push crackme2.40301F              40301F:"M}z{ji`}lfah0/Mnk/xnv."
0040124E    E8 C8020000         call crackme2.40151B
00401253    E9 B4000000         jmp crackme2.40130C
00401258    6A 0C               push C
0040125A    68 5D304000         push crackme2.40305D
0040125F    6A 6B               push 6B
```

Step 6: Analyze the Password Validation Logic

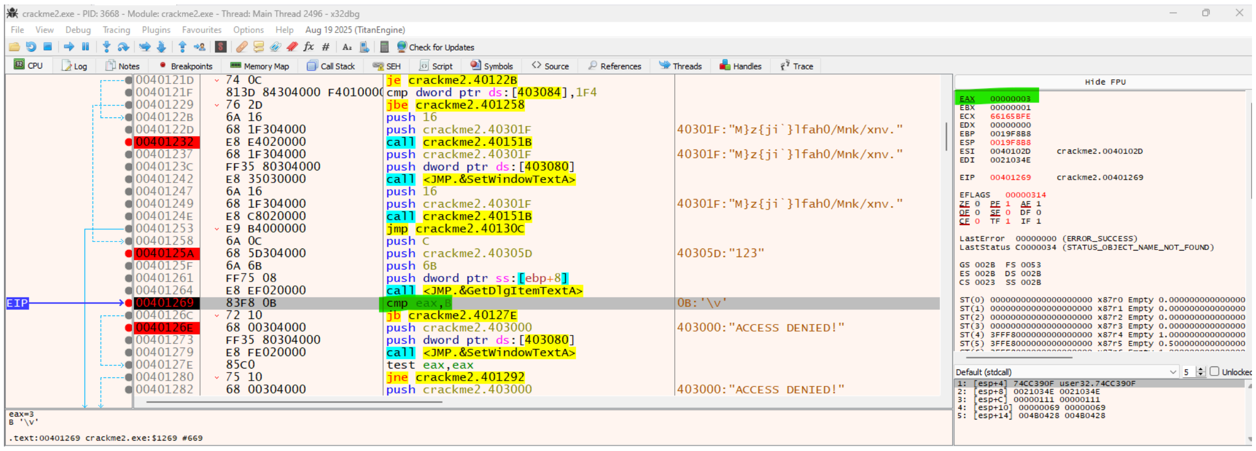

crackme2.exe - PID: 208 - Module: crackme2.exe - Thread: Main Thread 5352 - x32dbg
File View Debug Tracing Plugins Favourites Options Help Aug 19 2025 (TitanEngine)
Check for Updates
CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads Handles Trace
EIP

```
0040125A  68 5D304000   push crackme2.40305D                 40305D:"123"
0040125F  6A 6B         push 6B
00401261  FF75 08       push dword ptr ss:[ebp+8]
00401264  E8 EF020000   call <JMP.&GetDlgItemTextA>
00401269  83F8 0B       cmp eax,B                            0B:'\v'
0040126C  72 10         jb crackme2.40127E
0040126E  68 00304000   push crackme2.403000                 403000:"ACCESS DENIED!"
00401273  FF35 80304000 push dword ptr ds:[403080]
00401279  E8 FE020000   call <JMP.&SetWindowTextA>
0040127E  85C0          test eax,eax
00401280  75 10         jne crackme2.401292
00401282  68 00304000   push crackme2.403000                 403000:"ACCESS DENIED!"
00401287  FF35 80304000 push dword ptr ds:[403080]
0040128D  E8 EA020000   call <JMP.&SetWindowTextA>
00401292  50            push eax
00401293  68 5D304000   push crackme2.40305D                 40305D:"123"
00401298  E8 84010000   call crackme2.401421
0040129D  0BC0          or eax,eax
0040129F  75 1F         jne crackme2.4012C0
004012A1  68 0F304000   push crackme2.40300F                 40300F:"ACCESS GRANTED!"
004012A6  FF35 80304000 push dword ptr ds:[403080]
004012AC  E8 CB020000   call <JMP.&SetWindowTextA>
004012B1  6A 00         push 0
004012B3  FF35 80304000 push dword ptr ds:[403080]
004012B9  E8 88020000   call <JMP.&EnableWindow>
004012BE  EB 10         jmp crackme2.4012D0
```

Once the initial strings are decoded, a comparison operation is performed, checking the character count of what the user inputted as the password with the hex value ‘B’, 11 in decimal


crackme2.exe - PID: 3668 - Module: crackme2.exe - Thread: Main Thread 2496 - x32dbg
File View Debug Tracing Plugins Favourites Options Help Aug 19 2025 (TitanEngine)
Check for Updates
CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads Handles Trace
EIP

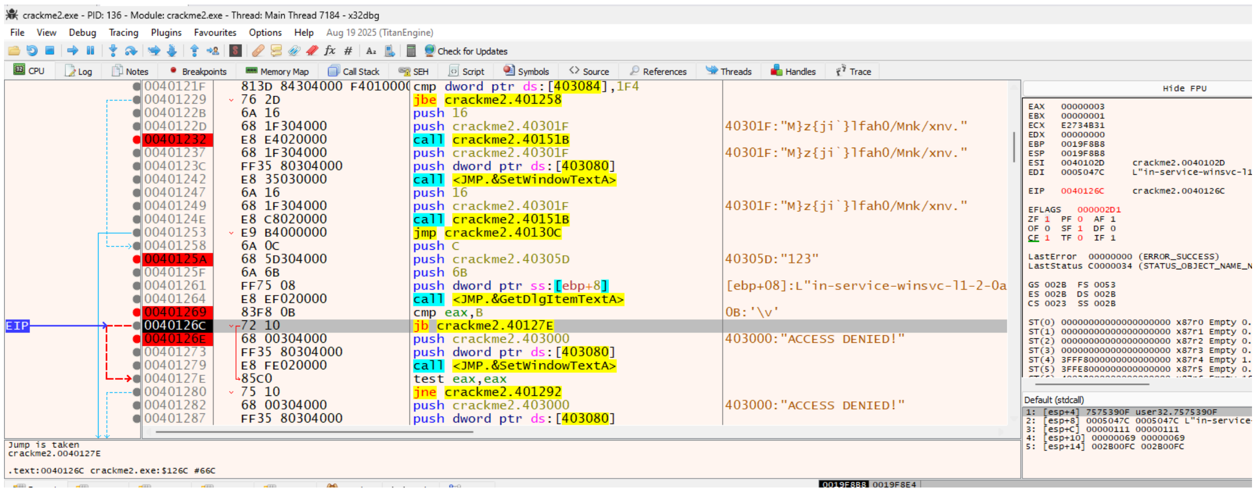
```
0040121D  74 0C                   je crackme2.40122B
0040121F  813D 84304000 F401000   cmp dword ptr ds:[403084],1F4
00401229  76 2D                   jbe crackme2.401258
0040122B  6A 16                   push 16
0040122D  68 1F304000             push crackme2.40301F       40301F:"M}z{ji`}lfah0/Mnk/xnv."
00401232  E8 E4020000             call crackme2.40151B
00401237  68 1F304000             push crackme2.40301F       40301F:"M}z{ji`}lfah0/Mnk/xnv."
0040123C  FF35 80304000           push dword ptr ds:[403080]
00401242  E8 35030000             call <JMP.&SetWindowTextA>
00401247  6A 16                   push 16
00401249  68 1F304000             push crackme2.40301F       40301F:"M}z{ji`}lfah0/Mnk/xnv."
0040124E  E8 C8020000             call crackme2.40151B
00401253  E9 B4000000             jmp crackme2.40130C
00401258  6A 0C                   push C
0040125A  68 5D304000             push crackme2.40305D       40305D:"123"
0040125F  6A 6B                   push 6B
00401261  FF75 08                 push dword ptr ss:[ebp+8]
00401264  E8 EF020000             call <JMP.&GetDlgItemTextA>
00401269  83F8 0B                 cmp eax,B                  0B:'\v'
0040126C  72 10                   jb crackme2.40127E
0040126E  68 00304000             push crackme2.403000       403000:"ACCESS DENIED!"
00401273  FF35 80304000           push dword ptr ds:[403080]
00401279  E8 FE020000             call <JMP.&SetWindowTextA>
0040127E  85C0                    test eax,eax
00401280  75 10                   jne crackme2.401292
00401282  68 00304000             push crackme2.403000       403000:"ACCESS DENIED!"
```



```
crackme2.exe - PID: 3668 - Module: crackme2.exe - Thread: Main Thread 2496 - x32dbg

0040121D  74 0C                 je crackme2.40122B
0040121F  813D 84304000 F401000 cmp dword ptr ds:[403084],1F4
00401229  76 2D                 jbe crackme2.401258
0040122B  6A 16                 push 16
0040122D  68 1F304000           push crackme2.40301F          40301F:"M}z{ji`}lfah0/Mnk/xnv."
00401232  E8 E4020000           call crackme2.40151B
00401237  68 1F304000           push crackme2.40301F          40301F:"M}z{ji`}lfah0/Mnk/xnv."
0040123C  FF35 80304000         push dword ptr ds:[403080]
00401242  E8 35030000           call <JMP.&SetWindowTextA>
00401247  6A 16                 push 16
00401249  68 1F304000           push crackme2.40301F          40301F:"M}z{ji`}lfah0/Mnk/xnv."
0040124E  E8 C8020000           call crackme2.40151B
00401253  E9 B4000000           jmp crackme2.40130C
00401258  6A 0C                 push C
0040125A  68 5D304000           push crackme2.40305D          40305D:"123"
0040125F  6A 6B                 push 6B
00401261  FF75 08               push dword ptr ss:[ebp+8]
00401264  E8 EF020000           call <JMP.&GetDlgItemTextA>
EIP 00401269  83F8 0B           cmp eax,B                     0B:'\v'
0040126C  72 10                 jb crackme2.40127E
0040126E  68 00304000           push crackme2.403000          403000:"ACCESS DENIED!"
00401273  FF35 80304000         push dword ptr ds:[403080]
00401279  E8 FE020000           call <JMP.&SetWindowTextA>
0040127E  85C0                  test eax,eax
00401280  75 10                 jne crackme2.401292
00401282  68 00304000           push crackme2.403000          403000:"ACCESS DENIED!"

eax=3
B '\v'

.text:00401269 crackme2.exe:$1269 #669
```

```
EAX   00000003
EBX   00000001
ECX   66165BFE
EDX   00000000
EBP   0019F8B8
ESP   0019F8B8
ESI   0040102D     crackme2.0040102D
EDI   0021034E

EIP   00401269     crackme2.00401269

EFLAGS   00000314
ZF 0  PF 1  AF 1
OF 0  SF 0  DF 0
CF 0  TF 1  IF 1

LastError  00000000 (ERROR_SUCCESS)
LastStatus C0000034 (STATUS_OBJECT_NAME_NOT_FOUND)

GS 002B  FS 0053
ES 002B  DS 002B
CS 0023  SS 002B

ST(0) 00000000000000000000 x87r0 Empty 0.000000000000000
ST(1) 00000000000000000000 x87r1 Empty 0.000000000000000
ST(2) 00000000000000000000 x87r2 Empty 0.000000000000000
ST(3) 00000000000000000000 x87r3 Empty 0.000000000000000
ST(4) 3FFF8000000000000000 x87r4 Empty 1.000000000000000
ST(5) 3FFE8000000000000000 x87r5 Empty 0.500000000000000

Default (stdcall)
1: [esp+4] 74CC390F user32.74CC390F
2: [esp+8] 0021034E 0021034E
3: [esp+C] 00000111 00000111
4: [esp+10] 00000069 00000069
5: [esp+14] 004B0428 004B0428
```

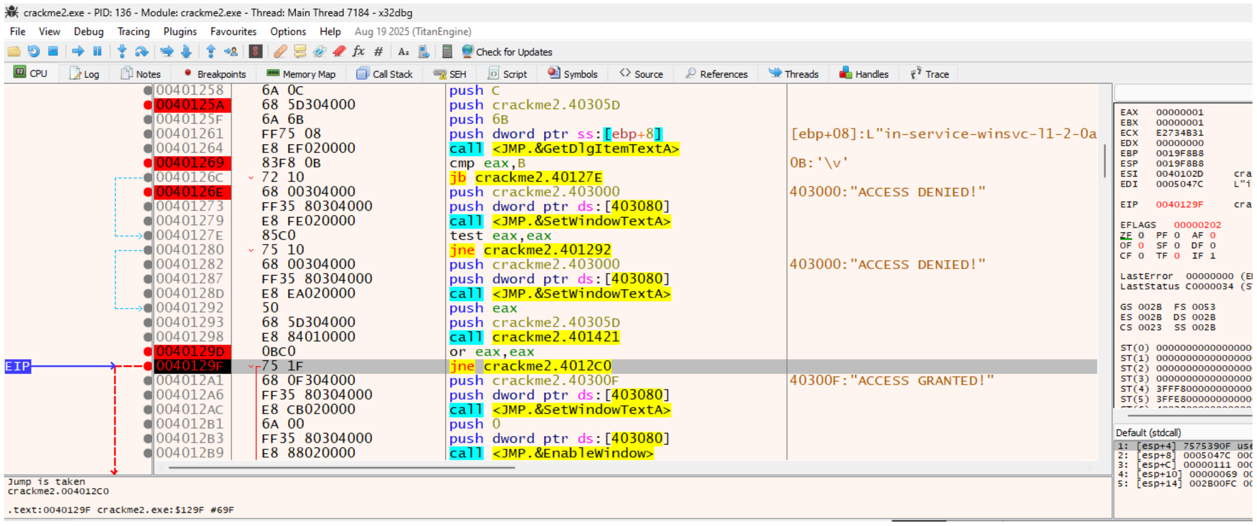
We can bypass this by changing the ZF to 1, where the code then thinks that the password passed the verification and then passes the Access Granted output onto the Dialog box



```
crackme2.exe - PID: 136 - Module: crackme2.exe - Thread: Main Thread 7184 - x32dbg

0040121F  813D 84304000 F401000 cmp dword ptr ds:[403084],1F4
00401229  76 2D                 jbe crackme2.401258
0040122B  6A 16                 push 16
0040122D  68 1F304000           push crackme2.40301F          40301F:"M}z{ji`}lfah0/Mnk/xnv."
00401232  E8 E4020000           call crackme2.40151B
00401237  68 1F304000           push crackme2.40301F          40301F:"M}z{ji`}lfah0/Mnk/xnv."
0040123C  FF35 80304000         push dword ptr ds:[403080]
00401242  E8 35030000           call <JMP.&SetWindowTextA>
00401247  6A 16                 push 16
00401249  68 1F304000           push crackme2.40301F          40301F:"M}z{ji`}lfah0/Mnk/xnv."
0040124E  E8 C8020000           call crackme2.40151B
00401253  E9 B4000000           jmp crackme2.40130C
00401258  6A 0C                 push C
0040125A  68 5D304000           push crackme2.40305D          40305D:"123"
0040125F  6A 6B                 push 6B
00401261  FF75 08               push dword ptr ss:[ebp+8]     [ebp+08]:L"in-service-winsvc-l1-2-0a
00401264  E8 EF020000           call <JMP.&GetDlgItemTextA>
00401269  83F8 0B               cmp eax,B                     0B:'\v'
EIP 0040126C  72 10             jb crackme2.40127E
0040126E  68 00304000           push crackme2.403000          403000:"ACCESS DENIED!"
00401273  FF35 80304000         push dword ptr ds:[403080]
00401279  E8 FE020000           call <JMP.&SetWindowTextA>
0040127E  85C0                  test eax,eax
00401280  75 10                 jne crackme2.401292
00401282  68 00304000           push crackme2.403000          403000:"ACCESS DENIED!"
00401287  FF35 80304000         push dword ptr ds:[403080]

Jump is taken
crackme2.0040127E

.text:0040126C crackme2.exe:$126C #66C
```

```
EAX   00000003
EBX   00000001
ECX   E2734B31
EDX   00000000
EBP   0019F8B8
ESP   0019F8B8
ESI   0040102D     crackme2.0040102D
EDI   0005047C     L"in-service-winsvc-l1

EIP   0040126C     crackme2.0040126C

EFLAGS   000002D1
ZF 1  PF 0  AF 1
OF 0  SF 1  DF 0
CF 1  TF 0  IF 1

LastError  00000000 (ERROR_SUCCESS)
LastStatus C0000034 (STATUS_OBJECT_NAME_N

GS 002B  FS 0053
ES 002B  DS 002B
CS 0023  SS 002B

ST(0) 00000000000000000000 x87r0 Empty 0.
ST(1) 00000000000000000000 x87r1 Empty 0.
ST(2) 00000000000000000000 x87r2 Empty 0.
ST(3) 00000000000000000000 x87r3 Empty 0.
ST(4) 3FFF8000000000000000 x87r4 Empty 1.
ST(5) 3FFE8000000000000000 x87r5 Empty 0.

Default (stdcall)
1: [esp+4] 7575390F user32.7575390F
2: [esp+8] 0005047C 0005047C L"in-service
3: [esp+C] 00000111 00000111
4: [esp+10] 00000069 00000069
5: [esp+14] 002B00FC 002B00FC
```

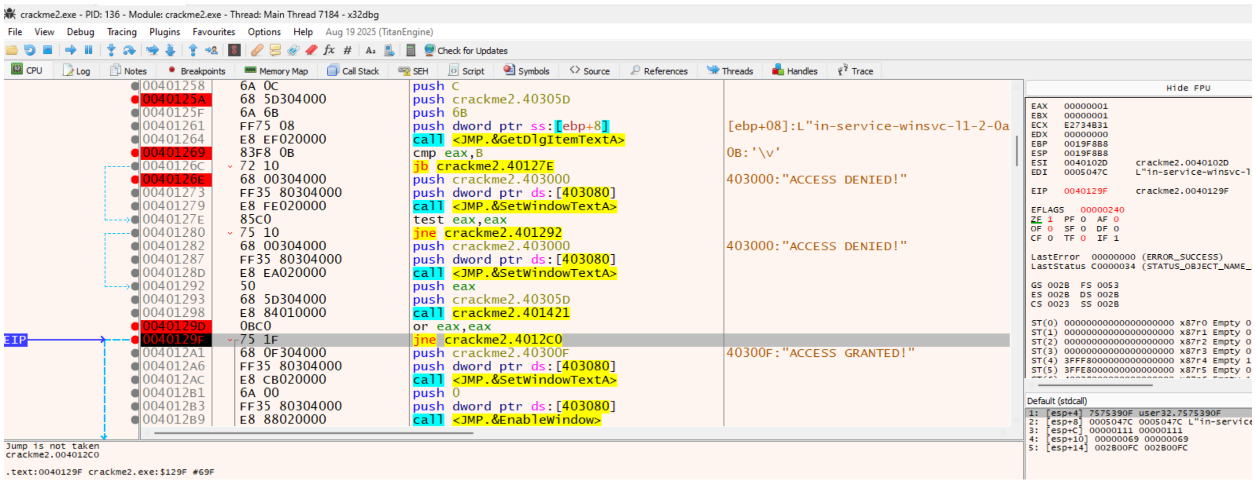
Next, we see the JNE instruction, which will determine which sentence will be printed


crackme2.exe - PID: 136 - Module: crackme2.exe - Thread: Main Thread 7184 - x32dbg
File View Debug Tracing Plugins Favourites Options Help Aug 19 2025 (TitanEngine)
Check for Updates
CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads Handles Trace

```
00401258   6A 0C          push C
0040125A   68 5D304000    push crackme2.40305D
0040125F   6A 6B          push 6B
00401261   FF75 08        push dword ptr ss:[ebp+8]             [ebp+08]:L"in-service-winsvc-l1-2-0a
00401264   E8 EF020000    call <JMP.&GetDlgItemTextA>
00401269   83F8 0B        cmp eax,B                             0B:'\v'
0040126C   72 10          jb crackme2.40127E
0040126E   68 00304000    push crackme2.403000                  403000:"ACCESS DENIED!"
00401273   FF35 80304000  push dword ptr ds:[403080]
00401279   E8 FE020000    call <JMP.&SetWindowTextA>
0040127E   85C0           test eax,eax
00401280   75 10          jne crackme2.401292
00401282   68 00304000    push crackme2.403000                  403000:"ACCESS DENIED!"
00401287   FF35 80304000  push dword ptr ds:[403080]
0040128D   E8 EA020000    call <JMP.&SetWindowTextA>
00401292   50             push eax
00401293   68 5D304000    push crackme2.40305D
00401298   E8 84010000    call crackme2.401421
0040129D   0BC0           or eax,eax
EIP 0040129F   75 1F      jne crackme2.4012C0
004012A1   68 0F304000    push crackme2.40300F                  40300F:"ACCESS GRANTED!"
004012A6   FF35 80304000  push dword ptr ds:[403080]
004012AC   E8 CB020000    call <JMP.&SetWindowTextA>
004012B1   6A 00          push 0
004012B3   FF35 80304000  push dword ptr ds:[403080]
004012B9   E8 88020000    call <JMP.&EnableWindow>

Jump is taken
crackme2.004012C0

.text:0040129F crackme2.exe:$129F #69F
```

```
EAX   00000001
EBX   00000001
ECX   E2734B31
EDX   00000000
EBP   0019F8B8
ESP   0019F8B8
ESI   0040102D      cra
EDI   0005047C      L"i

EIP   0040129F      cra

EFLAGS   00000202
ZF 0  PF 0  AF 0
OF 0  SF 0  DF 0
CF 0  TF 0  IF 1

LastError  00000000 (E
LastStatus C0000034 (S

GS 002B  FS 0053
ES 002B  DS 002B
CS 0023  SS 002B

ST(0) 0000000000000000
ST(1) 0000000000000000
ST(2) 0000000000000000
ST(3) 0000000000000000
ST(4) 3FFF800000000000
ST(5) 3FFE800000000000

Default (stdcall)
1: [esp+4] 7575390F use
2: [esp+8] 0005047C 000
3: [esp+C] 00000111 000
4: [esp+10] 00000069 00
5: [esp+14] 002B00FC 00
```

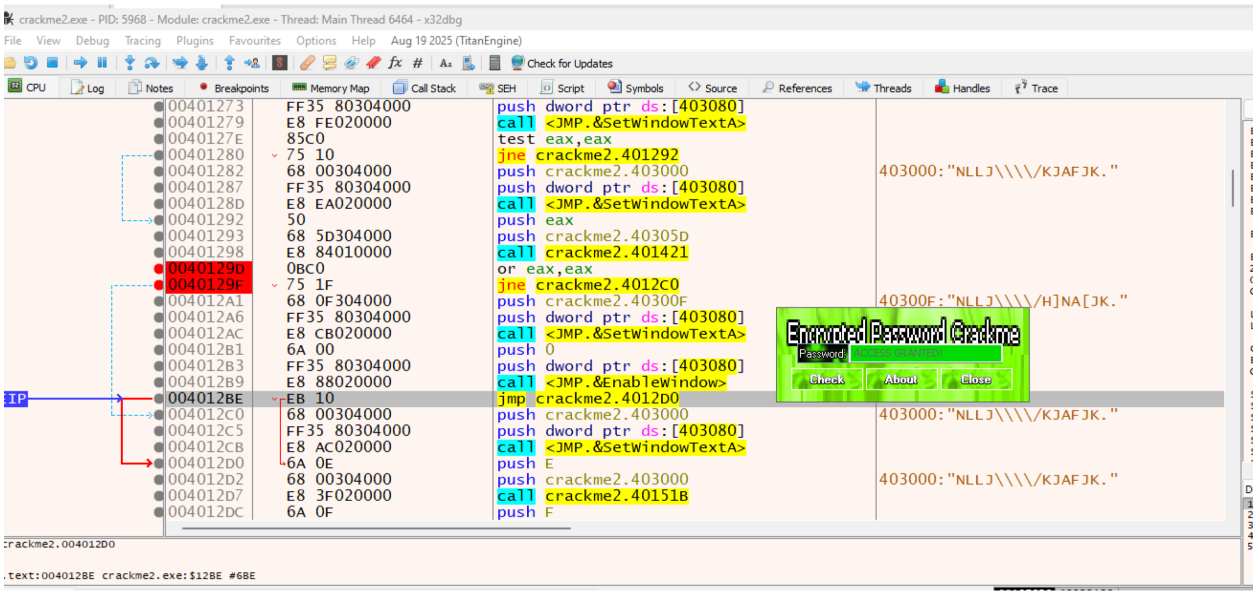
Same step, change the ZF to 1 to prevent the jump


crackme2.exe - PID: 136 - Module: crackme2.exe - Thread: Main Thread 7184 - x32dbg
File View Debug Tracing Plugins Favourites Options Help Aug 19 2025 (TitanEngine)
Check for Updates
CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads Handles Trace
Hide FPU

```
00401258   6A 0C          push C
0040125A   68 5D304000    push crackme2.40305D
0040125F   6A 6B          push 6B
00401261   FF75 08        push dword ptr ss:[ebp+8]             [ebp+08]:L"in-service-winsvc-l1-2-0a
00401264   E8 EF020000    call <JMP.&GetDlgItemTextA>
00401269   83F8 0B        cmp eax,B                             0B:'\v'
0040126C   72 10          jb crackme2.40127E
0040126E   68 00304000    push crackme2.403000                  403000:"ACCESS DENIED!"
00401273   FF35 80304000  push dword ptr ds:[403080]
00401279   E8 FE020000    call <JMP.&SetWindowTextA>
0040127E   85C0           test eax,eax
00401280   75 10          jne crackme2.401292
00401282   68 00304000    push crackme2.403000                  403000:"ACCESS DENIED!"
00401287   FF35 80304000  push dword ptr ds:[403080]
0040128D   E8 EA020000    call <JMP.&SetWindowTextA>
00401292   50             push eax
00401293   68 5D304000    push crackme2.40305D
00401298   E8 84010000    call crackme2.401421
0040129D   0BC0           or eax,eax
EIP 0040129F   75 1F      jne crackme2.4012C0
004012A1   68 0F304000    push crackme2.40300F                  40300F:"ACCESS GRANTED!"
004012A6   FF35 80304000  push dword ptr ds:[403080]
004012AC   E8 CB020000    call <JMP.&SetWindowTextA>
004012B1   6A 00          push 0
004012B3   FF35 80304000  push dword ptr ds:[403080]
004012B9   E8 88020000    call <JMP.&EnableWindow>

Jump is not taken
crackme2.004012C0

.text:0040129F crackme2.exe:$129F #69F
```

```
EAX   00000001
EBX   00000001
ECX   E2734B31
EDX   00000000
EBP   0019F8B8
ESP   0019F8B8
ESI   0040102D      crackme2.0040102D
EDI   0005047C      L"in-service-winsvc-l

EIP   0040129F      crackme2.0040129F

EFLAGS   00000240
ZF 1  PF 0  AF 0
OF 0  SF 0  DF 0
CF 0  TF 0  IF 1

LastError  00000000 (ERROR_SUCCESS)
LastStatus C0000034 (STATUS_OBJECT_NAME_

GS 002B  FS 0053
ES 002B  DS 002B
CS 0023  SS 002B

ST(0) 00000000000000000000 x87r0 Empty 0
ST(1) 00000000000000000000 x87r1 Empty 0
ST(2) 00000000000000000000 x87r2 Empty 0
ST(3) 00000000000000000000 x87r3 Empty 0
ST(4) 3FFF8000000000000000 x87r4 Empty 1
ST(5) 3FFE8000000000000000 x87r5 Empty 0

Default (stdcall)
1: [esp+4] 7575390F user32.7575390F
2: [esp+8] 0005047C 0005047C L"in-service
3: [esp+C] 00000111 00000111
4: [esp+10] 00000069 00000069
5: [esp+14] 002B00FC 002B00FC
```

At this point, click the run again, and you will see the message ACCESS GRANTED.


crackme2.exe - PID: 5968 - Module: crackme2.exe - Thread: Main Thread 6464 - x32dbg
File View Debug Tracing Plugins Favourites Options Help Aug 19 2025 (TitanEngine)
Check for Updates
CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads Handles Trace
EIP
Encrypted Password Crackme
Password:
Check
About
Close

```
00401273   FF35 80304000   push dword ptr ds:[403080]
00401279   E8 FE020000     call <JMP.&SetWindowTextA>
0040127E   85C0            test eax,eax
00401280   75 10           jne crackme2.401292
00401282   68 00304000     push crackme2.403000            403000:"NLLJ\\\\/KJAFJK."
00401287   FF35 80304000   push dword ptr ds:[403080]
0040128D   E8 EA020000     call <JMP.&SetWindowTextA>
00401292   50              push eax
00401293   68 5D304000     push crackme2.40305D
00401298   E8 84010000     call crackme2.401421
0040129D   0BC0            or eax,eax
0040129F   75 1F           jne crackme2.4012C0
004012A1   68 0F304000     push crackme2.40300F            40300F:"NLLJ\\\\/H]NA[JK."
004012A6   FF35 80304000   push dword ptr ds:[403080]
004012AC   E8 CB020000     call <JMP.&SetWindowTextA>
004012B1   6A 00           push 0
004012B3   FF35 80304000   push dword ptr ds:[403080]
004012B9   E8 88020000     call <JMP.&EnableWindow>
004012BE   EB 10           jmp crackme2.4012D0
004012C0   68 00304000     push crackme2.403000            403000:"NLLJ\\\\/KJAFJK."
004012C5   FF35 80304000   push dword ptr ds:[403080]
004012CB   E8 AC020000     call <JMP.&SetWindowTextA>
004012D0   6A 0E           push E
004012D2   68 00304000     push crackme2.403000            403000:"NLLJ\\\\/KJAFJK."
004012D7   E8 3F020000     call crackme2.40151B
004012DC   6A 0F           push F
```

ACCESS GRANTED!

```
crackme2.004012D0

.text:004012BE crackme2.exe:$12BE #6BE
```