Requirements: OllyDbg

Open the crackme with OllyDbg, go to 0x00402506 which represent the onLoad handler, go down,and stop at the address 0x00402606.

```
00402609  .  50              PUSH EAX
0040260A  .  68 8C204000     PUSH 0040208C                            UNICODE ".exe"
0040260F  .  FF15 2810400(   CALL DWORD PTR DS:[<&MSVBVM60.__vbaStrCat>]
00402615  .  8BD0            MOV EDX,EAX
00402617  .  8D4D E8         LEA ECX,[LOCAL.6]
0040261A  .  FF15 A810400(   CALL DWORD PTR DS:[<&MSVBVM60.__vbaStrMove>]
00402620  .  8D4D E4         LEA ECX,[LOCAL.7]
00402623  .  FF15 BC10400(   CALL DWORD PTR DS:[<&MSVBVM60.__vbaFreeStr>]
00402629  .  8D4D E0         LEA ECX,[LOCAL.8]
0040262C  .  FF15 C010400(   CALL DWORD PTR DS:[<&MSVBVM60.__vbaFreeObj>]
00402632  .  8B4D E8         MOV ECX,DWORD PTR SS:[LOCAL.6]
00402635  .  51              PUSH ECX
00402636  .  68 9C204000     PUSH 0040209C                            UNICODE "Crackme.exe"
0040263B  .  FF15 5410400(   CALL DWORD PTR DS:[<&MSVBVM60.__vbaStrCmp>]
00402641  .  85C0            TEST EAX,EAX
00402643  .v 0F84 8000000(   JE 004026C9
```

Has you can see, in this snippet of code, the crackme will compare the actual name of the crackme, with “Crackme.exe”, and if they are different we can't go forward, so don't rename the crackme. Next we will reach another validation which uses the api “IsDebuggerPresent” to avoid debug.

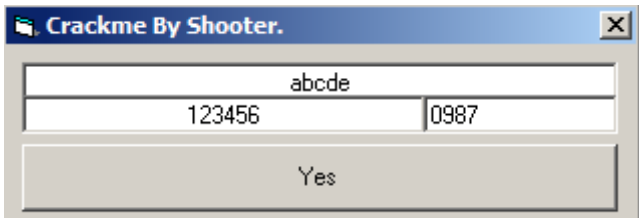
```
004026C9  >  8B35 9C10400(   MOV ESI,DWORD PTR DS:[<&MSVBVM60.__vbaVarDup>]
004026CF  .  BB 0A000000     MOV EBX,0A
004026D4  .  BF 08000000     MOV EDI,8
004026D9  >  E8 16F9FFFF     CALL 00401FF4
```

Step into 0x004026D9, a you will see IsDebuggerPresent api call.

```
00401FF4  $  A1 B0334000     MOV EAX,DWORD PTR DS:[4033B0]            Jump to KERNELBASE.IsDebuggerPresent
00401FF9  .  0BC0            OR EAX,EAX
00401FFB  .v 74 02           JE SHORT 00401FFF
00401FFD  .  FFE0            JMP EAX
00401FFF  >  68 DC1F4000     PUSH 00401FDC                            PTR to ASCII "kernel32.dll"
00402004  .  B8 B0114000     MOV EAX,<JMP.&MSVBVM60.DllFunctionCall>  Jump to MSVBVM60.DllFunctionCall
00402009  .  FFD0            CALL EAX
0040200B  .  FFE0            JMP EAX                                  kernel32.IsDebuggerPresent
```

So just, patch the JNE located at the address 00x4026F1 with a JMP, and resume the process.

Set your data like this!



So, now we just have to locate the “Yes” button click handler, and get the correct serial. Set a breakpoint at the address 0x004027C0 and press on the “Yes” button.

```
00402966  .  6A 03           PUSH 3
00402968  .  8D4D A4         LEA ECX,[LOCAL.23]
0040296B  .  51              PUSH ECX
0040296C  .  8D55 B4         LEA EDX,[LOCAL.19]
0040296F  .  8D45 D8         LEA EAX,[LOCAL.10]
00402972  .  52              PUSH EDX
00402973  .  8945 AC         MOV DWORD PTR SS:[LOCAL.21],EAX
00402976  .  C745 A4 0840(   MOV DWORD PTR SS:[LOCAL.23],4008
00402979  .  FF15 A010400(   CALL DWORD PTR DS:[<&MSVBVM60.#617>]
```

At the address 0x0040297D the crackme retrieve the first 3 chars our username, with the api “rtcLeftCharVar”.

```
004029A1  ·  6A 03          PUSH 3
004029A3  ·  8D55 A4        LEA EDX,[LOCAL.23]
004029A6  ·  52             PUSH EDX
004029A7  ·  8D45 B4        LEA EAX,[LOCAL.19]
004029AA  ·  8D4D D8        LEA ECX,[LOCAL.10]
004029AD  ·  50             PUSH EAX
004029AE  ·  894D AC        MOV DWORD PTR SS:[LOCAL.21],ECX
004029B1  ·  C745 A4 0840(  MOV DWORD PTR SS:[LOCAL.23],4008
004029B8  ·  FF15 AC10400(  CALL DWORD PTR DS:[<&MSVBVM60.#619>]
```

And at the address 0x004029B8 the crackme will call the api “rtcRightCharVal” to get the last 3 chars of our username.

```
004029D9  ·  68 D0214000    PUSH 004021D0                    UNICODE "SCT-"
004029DE  ·  52             PUSH EDX
004029DF  ·  FFD7           CALL EDI
004029E1  ·  8BD0           MOV EDX,EAX                      UNICODE "SCT-cde"
```

Next, our username last 3 chars will be added to the string “SCT-”, which means that one of our serials will be at this time “SCT-cde”.

```
004029E8  ·  50             PUSH EAX
004029E9  ·  8B45 D8        MOV EAX,DWORD PTR SS:[LOCAL.10]
004029EC  ·  50             PUSH EAX
004029ED  ·  FF15 1410400(  CALL DWORD PTR DS:[<&MSVBVM60.__vbaLenBstr>]
004029F3  ·  50             PUSH EAX                         get actual lenght
004029F4  ·  FF15 0810400(  CALL DWORD PTR DS:[<&MSVBVM60.__vbaStrI4>]
004029FA  ·  8BD0           MOV EDX,EAX
004029FC  ·  8D4D CC        LEA ECX,[LOCAL.13]
004029FF  ·  FFD6           CALL ESI
00402A01  ·  50             PUSH EAX
00402A02  ·  FFD7           CALL EDI
00402A04  ·  8BD0           MOV EDX,EAX                      UNICODE "SCT-cde5"
```

Next, the crackme will getn the length of our username, and add it to the serial, which is now “SCT-cde5”.

```
00402A0E  ·  50             PUSH EAX
00402A0F  ·  51             PUSH ECX
00402A10  ·  FFD7           CALL EDI
00402A12  ·  8BD0           MOV EDX,EAX                      UNICODE "SCT-cde5abc"
00402A14  ·  8D4D E8        LEA ECX,[LOCAL.6]                "SCT-cde5abc"
```

And at last the serial is finished by adding the first 3 chars to the end of it. “SCT-cde5abc”.

```
00402A3A  ·  68 E0214000    PUSH 004021E0                    UNICODE "Shooter"
00402A3F  ·  FFD7           CALL EDI
```

At the address 0x00402A37, the second serial will be compared to “Shooter”, so now we can assume that:

Username: abcde
Serial1: SCT-cde5abc
Serial2: Shooter

Go see the keygen source!