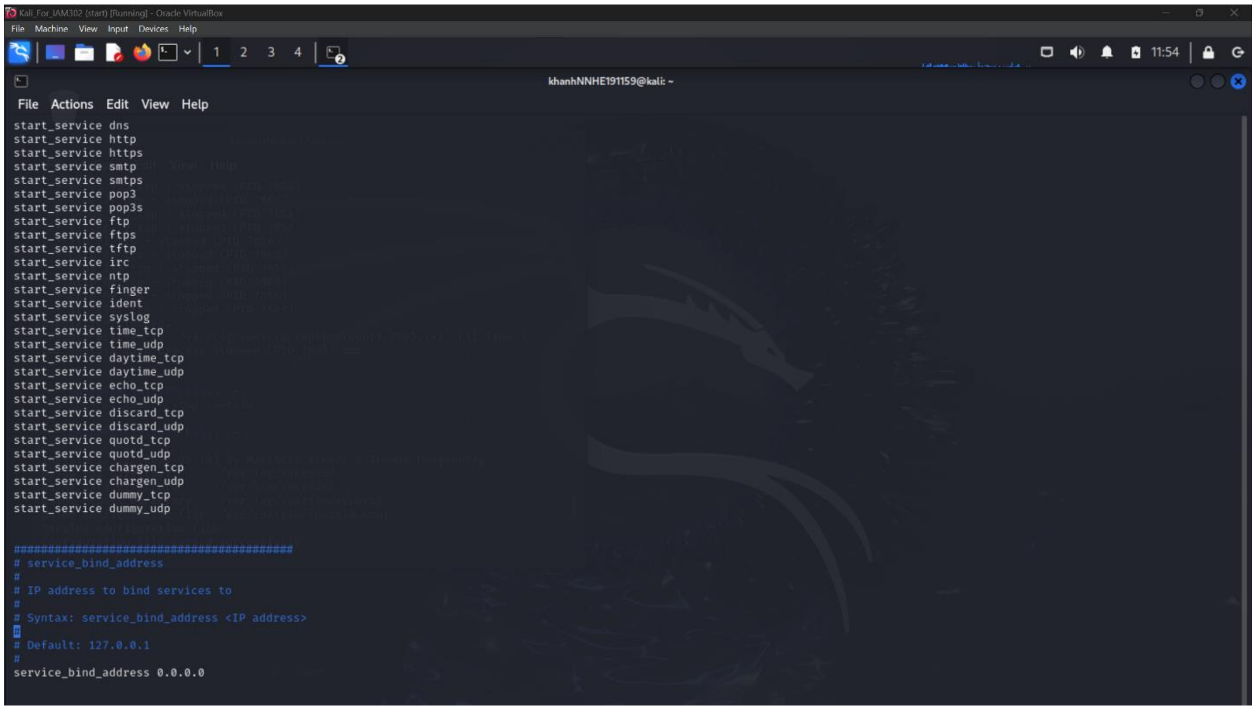
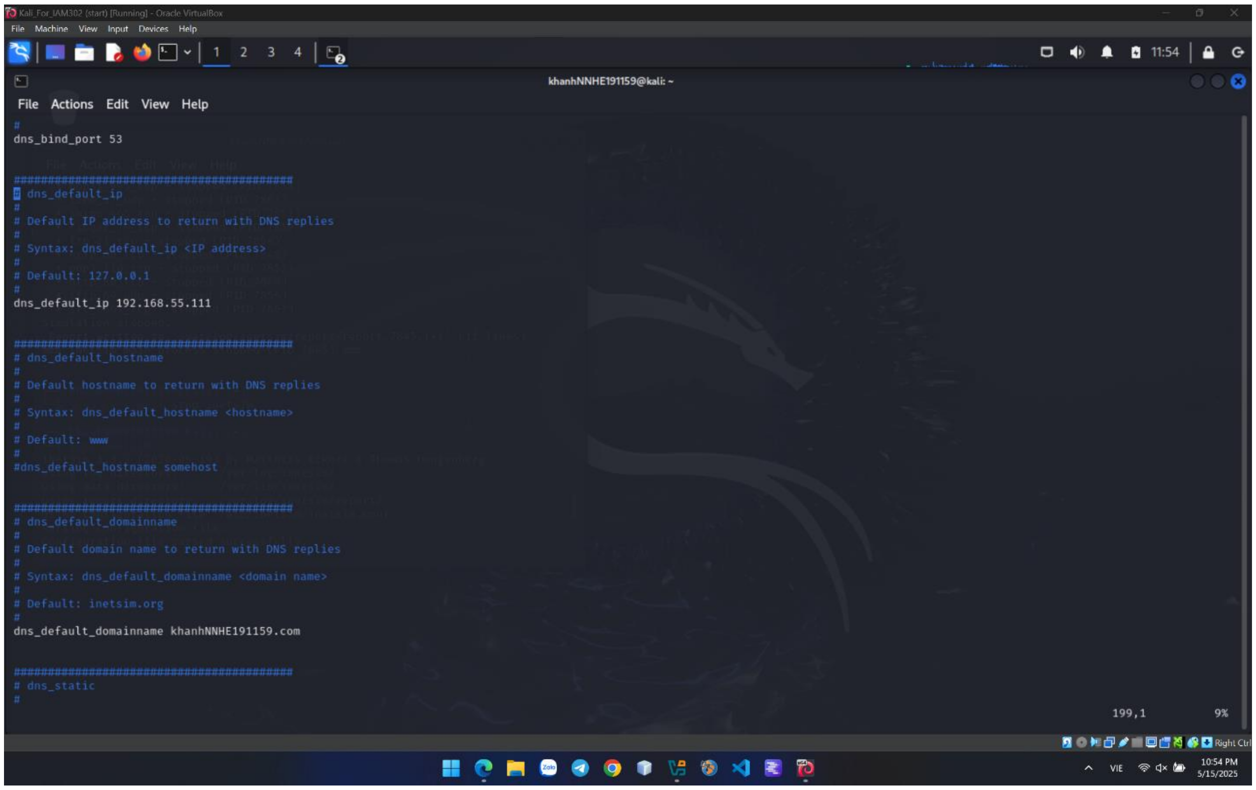
Nguyễn Nam Khánh – HE191159 – IA1902 – IAM302

LAB 1: Setting Up Environment

Setup Inetsim configuration for Kali

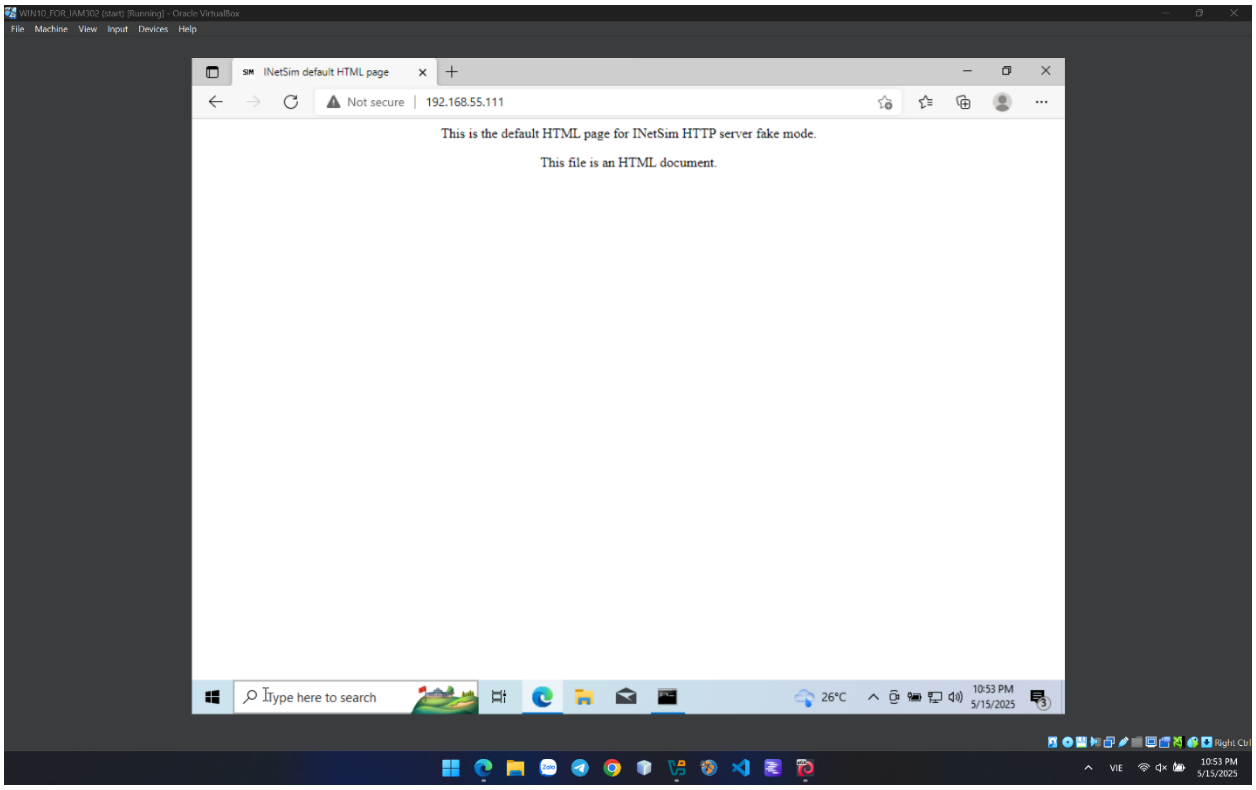


```
start_service dns
start_service http
start_service https
start_service smtp
start_service smtps
start_service pop3
start_service pop3s
start_service ftp
start_service ftps
start_service tftp
start_service irc
start_service ntp
start_service finger
start_service ident
start_service syslog
start_service time_tcp
start_service time_udp
start_service daytime_tcp
start_service daytime_udp
start_service echo_tcp
start_service echo_udp
start_service discard_tcp
start_service discard_udp
start_service quotd_tcp
start_service quotd_udp
start_service chargen_tcp
start_service chargen_udp
start_service dummy_tcp
start_service dummy_udp

#########################################
# service_bind_address
#
# IP address to bind services to
#
# Syntax: service_bind_address <IP address>
#
# Default: 127.0.0.1
#
service_bind_address 0.0.0.0
```



```
#
dns_bind_port 53

#########################################
# dns_default_ip
#
# Default IP address to return with DNS replies
#
# Syntax: dns_default_ip <IP address>
#
# Default: 127.0.0.1
#
dns_default_ip 192.168.55.111

#########################################
# dns_default_hostname
#
# Default hostname to return with DNS replies
#
# Syntax: dns_default_hostname <hostname>
#
# Default: www
#
#dns_default_hostname somehost

#########################################
# dns_default_domainname
#
# Default domain name to return with DNS replies
#
# Syntax: dns_default_domainname <domain name>
#
# Default: inetsim.org
#
dns_default_domainname khanhNNHE191159.com

#########################################
# dns_static
#
                                                          199,1         9%
```

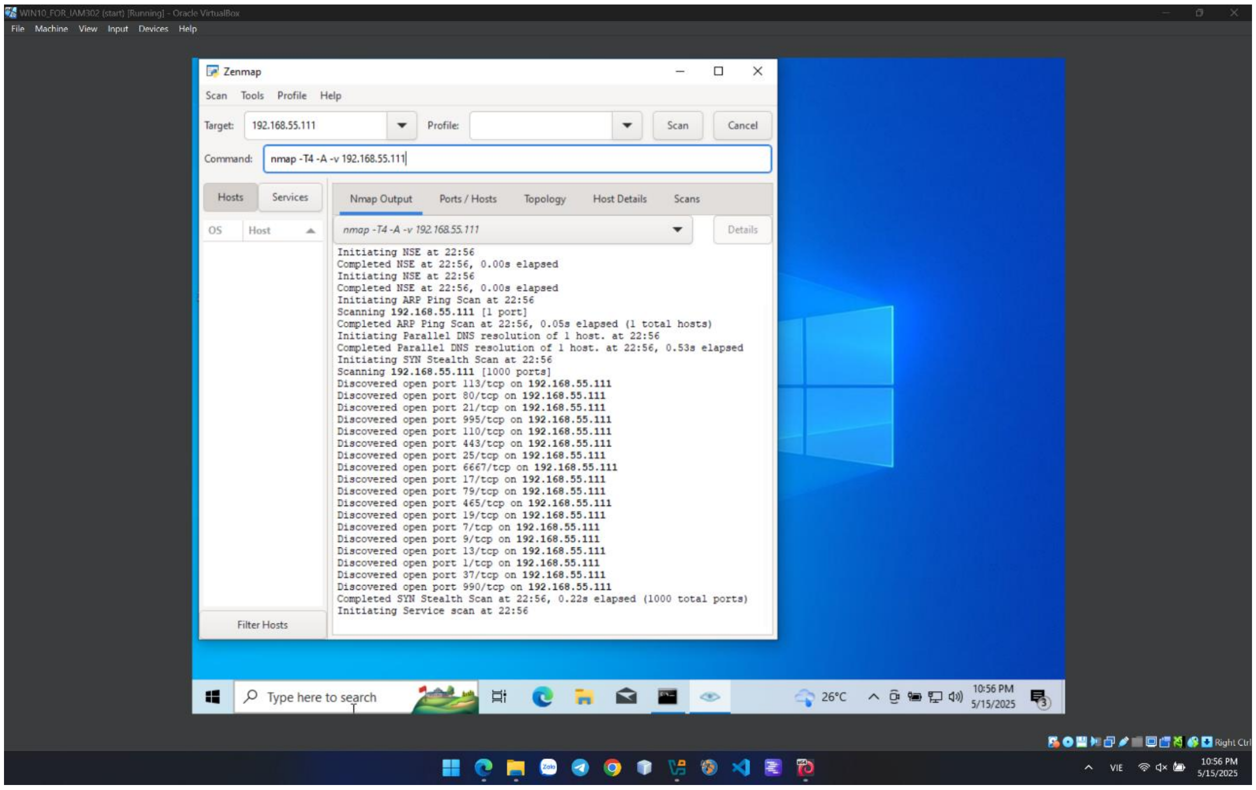
Viewing an HTTP Web Page



This is the default HTML page for INetSim HTTP server fake mode.

This file is an HTML document.

Start Nmap



Target: 192.168.55.111

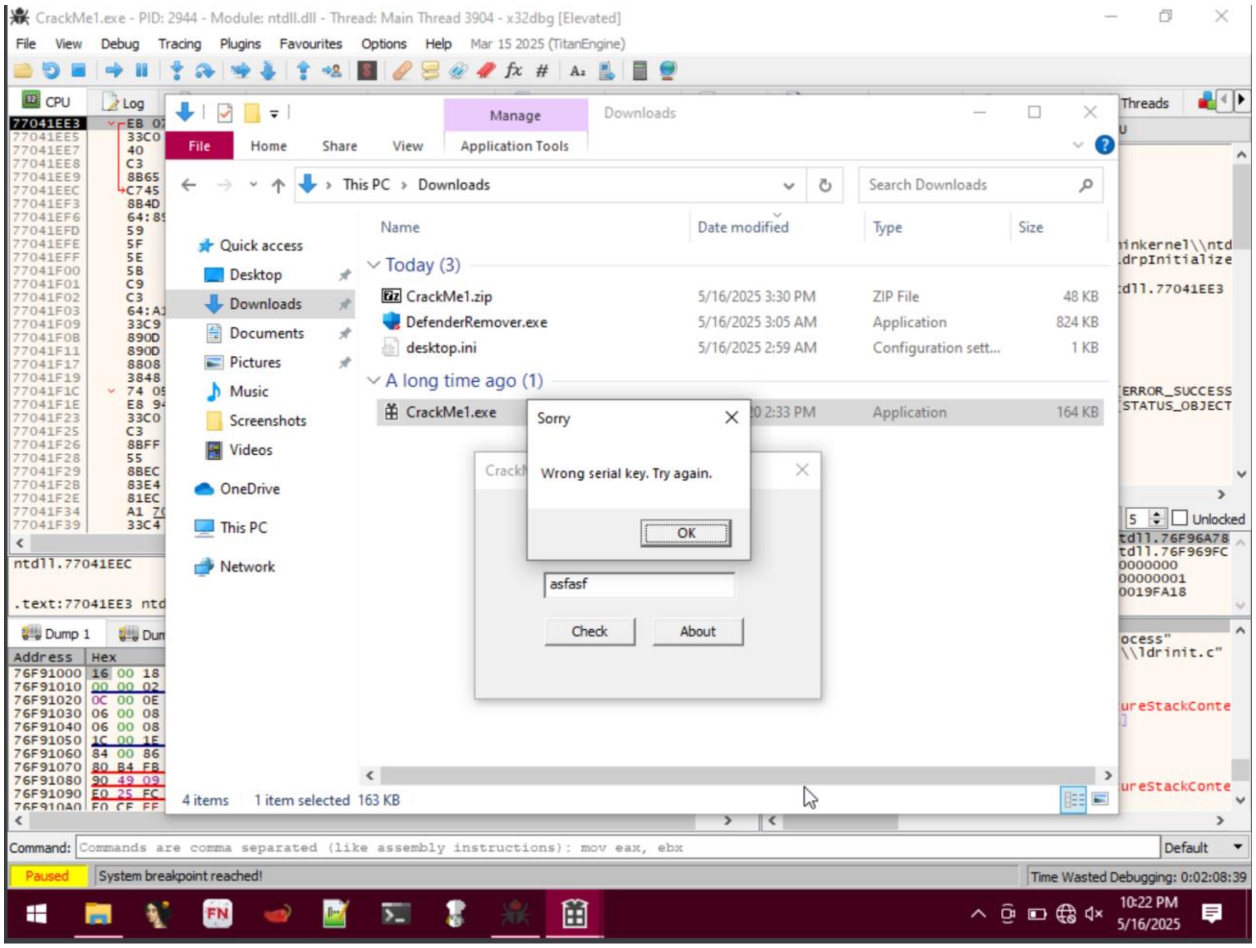
Command: nmap -T4 -A -v 192.168.55.111

```
Initiating NSE at 22:56
Completed NSE at 22:56, 0.00s elapsed
Initiating NSE at 22:56
Completed NSE at 22:56, 0.00s elapsed
Initiating ARP Ping Scan at 22:56
Scanning 192.168.55.111 [1 port]
Completed ARP Ping Scan at 22:56, 0.05s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 22:56
Completed Parallel DNS resolution of 1 host. at 22:56, 0.53s elapsed
Initiating SYN Stealth Scan at 22:56
Scanning 192.168.55.111 [1000 ports]
Discovered open port 113/tcp on 192.168.55.111
Discovered open port 80/tcp on 192.168.55.111
Discovered open port 21/tcp on 192.168.55.111
Discovered open port 995/tcp on 192.168.55.111
Discovered open port 110/tcp on 192.168.55.111
Discovered open port 443/tcp on 192.168.55.111
Discovered open port 25/tcp on 192.168.55.111
Discovered open port 6667/tcp on 192.168.55.111
Discovered open port 17/tcp on 192.168.55.111
Discovered open port 79/tcp on 192.168.55.111
Discovered open port 465/tcp on 192.168.55.111
Discovered open port 19/tcp on 192.168.55.111
Discovered open port 7/tcp on 192.168.55.111
Discovered open port 9/tcp on 192.168.55.111
Discovered open port 13/tcp on 192.168.55.111
Discovered open port 1/tcp on 192.168.55.111
Discovered open port 37/tcp on 192.168.55.111
Discovered open port 990/tcp on 192.168.55.111
Completed SYN Stealth Scan at 22:56, 0.22s elapsed (1000 total ports)
Initiating Service scan at 22:56
```

Nguyễn Nam Khánh – HE191159 – IA1902 – IAM302

BÀI CRACKME 1

Open the CrackMe1.exe file and test it with a random serial key, resulting in the message "Wrong serial key. Try again." This is a critical indicator, as it will guide us to locate the correct serial key in subsequent steps by analyzing the related code.



CrackMe1.exe - PID: 2944 - Module: ntdll.dll - Thread: Main Thread 3904 - x32dbg [Elevated]

Sorry

Wrong serial key. Try again.

OK

asfasf

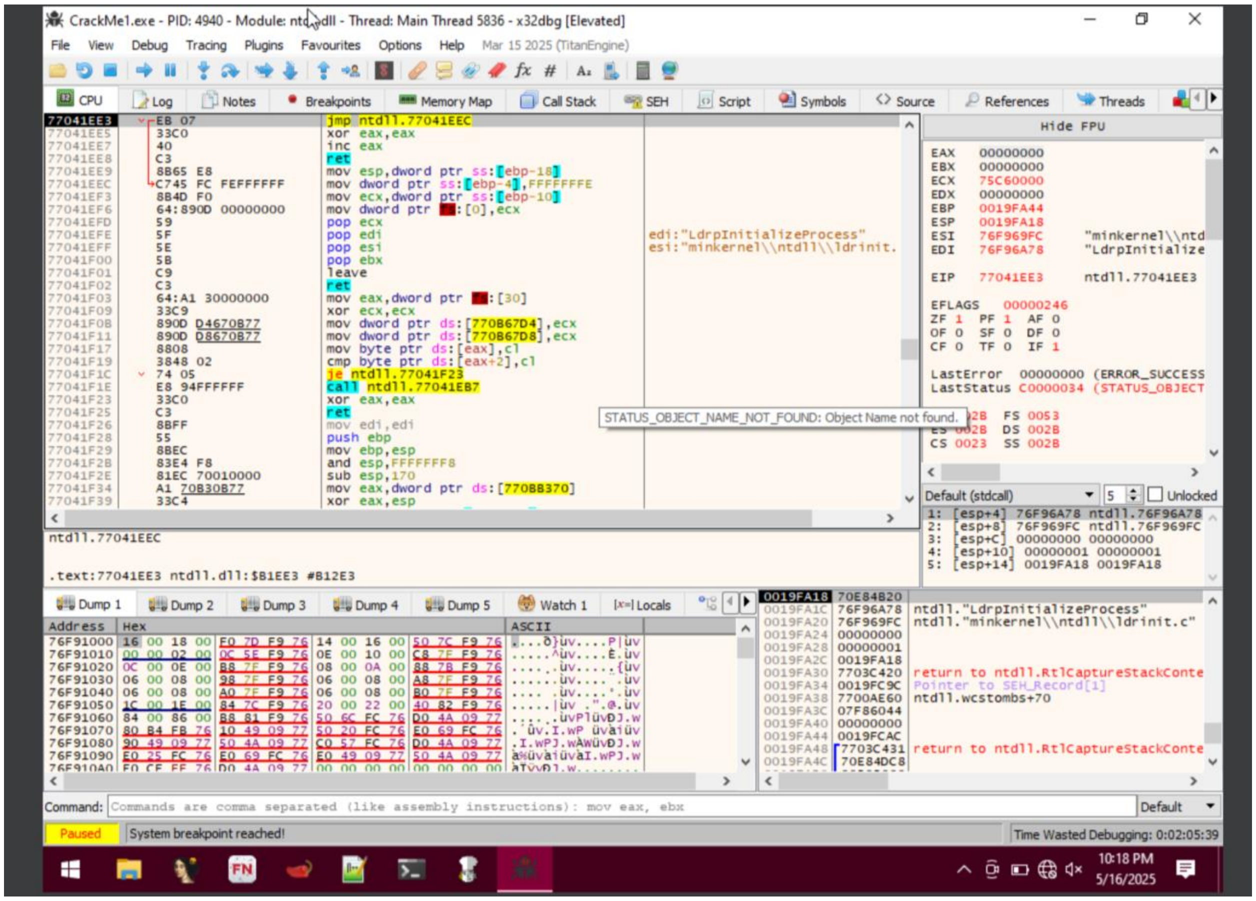
Check About

Command: Commands are comma separated (like assembly instructions): mov eax, ebx

Paused System breakpoint reached! Time Wasted Debugging: 0:02:08:39

Open the CrackMe1.exe file in x32dbg to start the analysis.

Note: x32dbg is a debugger used to analyze the program's machine code (assembly). The CPU tab displays the assembly instructions we will examine.


CrackMe1.exe - PID: 4940 - Module: ntdll - Thread: Main Thread 5836 - x32dbg [Elevated]
File View Debug Tracing Plugins Favourites Options Help Mar 15 2025 (TitanEngine)
CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads
Hide FPU
Dump 1 Dump 2 Dump 3 Dump 4 Dump 5 Watch 1 Locals
Default (stdcall) 5 Unlocked
Default

```
77041EE3   EB 07              jmp ntdll.77041EEC
77041EE5   33C0               xor eax,eax
77041EE7   40                 inc eax
77041EE8   C3                 ret
77041EE9   8B65 E8            mov esp,dword ptr ss:[ebp-18]
77041EEC   C745 FC FEFFFFFF   mov dword ptr ss:[ebp-4],FFFFFFFE
77041EF3   8B4D F0            mov ecx,dword ptr ss:[ebp-10]
77041EF6   64:890D 00000000   mov dword ptr fs:[0],ecx
77041EFD   59                 pop ecx
77041EFE   5F                 pop edi                              edi:"LdrpInitializeProcess"
77041EFF   5E                 pop esi                              esi:"minkernel\\ntdll\\ldrinit.
77041F00   5B                 pop ebx
77041F01   C9                 leave
77041F02   C3                 ret
77041F03   64:A1 30000000     mov eax,dword ptr fs:[30]
77041F09   33C9               xor ecx,ecx
77041F0B   890D D4670B77      mov dword ptr ds:[770B67D4],ecx
77041F11   890D D8670B77      mov dword ptr ds:[770B67D8],ecx
77041F17   8808               mov byte ptr ds:[eax],cl
77041F19   3848 02            cmp byte ptr ds:[eax+2],cl
77041F1C   74 05              je ntdll.77041F23
77041F1E   E8 94FFFFFF        call ntdll.77041EB7
77041F23   33C0               xor eax,eax
77041F25   C3                 ret
77041F26   8BFF               mov edi,edi
77041F28   55                 push ebp
77041F29   8BEC               mov ebp,esp
77041F2B   83E4 F8            and esp,FFFFFFF8
77041F2E   81EC 70010000      sub esp,170
77041F34   A1 70B30B77        mov eax,dword ptr ds:[770BB370]
77041F39   33C4               xor eax,esp

STATUS_OBJECT_NAME_NOT_FOUND: Object Name not found.

ntdll.77041EEC

.text:77041EE3 ntdll.dll:$B1EE3 #B12E3
```

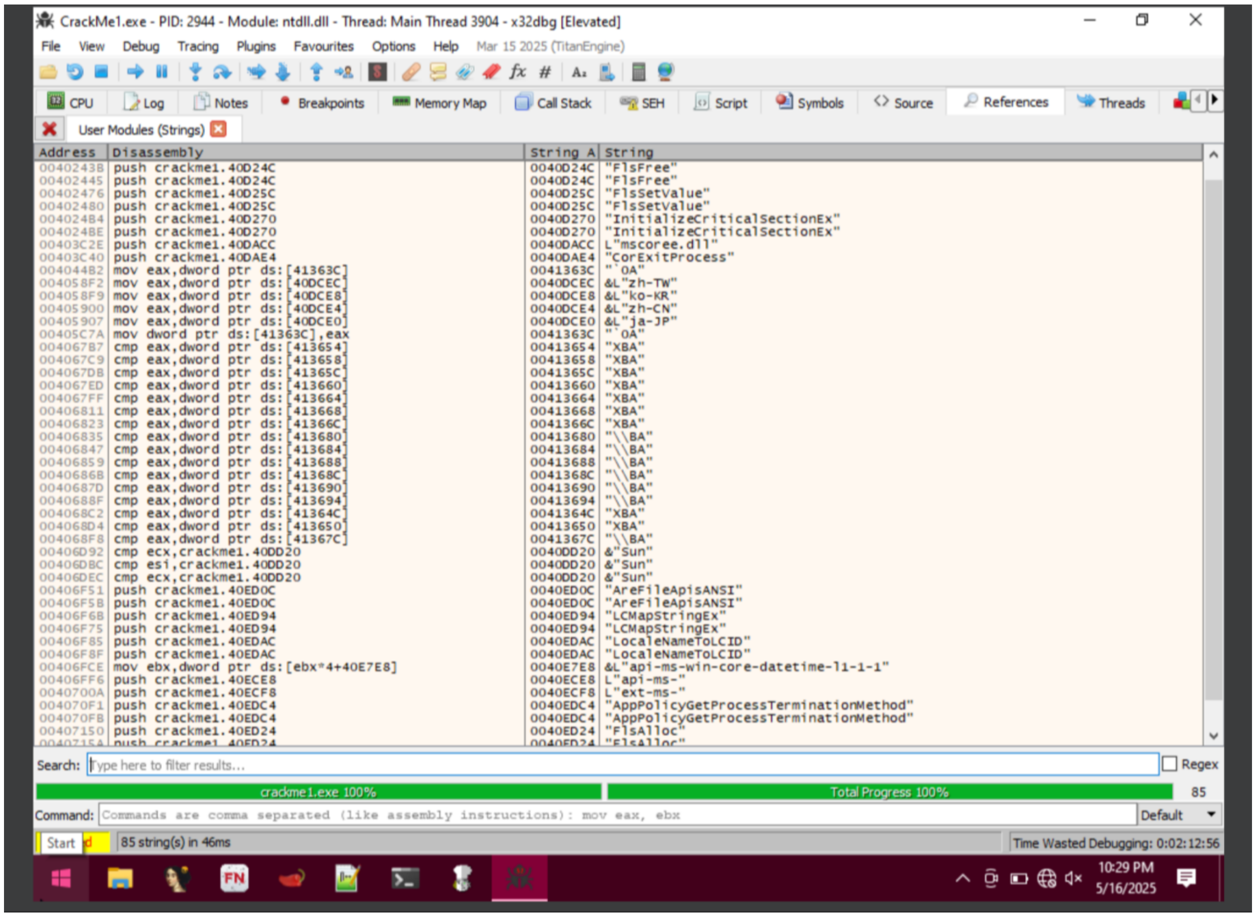
```
EAX   00000000
EBX   00000000
ECX   75C60000
EDX   00000000
EBP   0019FA44
ESP   0019FA18
ESI   76F969FC     "minkernel\\ntd
EDI   76F96A78     "LdrpInitialize

EIP   77041EE3     ntdll.77041EE3

EFLAGS   00000246
ZF 1  PF 1  AF 0
OF 0  SF 0  DF 0
CF 0  TF 0  IF 1

LastError   00000000 (ERROR_SUCCESS
LastStatus  C0000034 (STATUS_OBJECT

      2B  FS 0053
ES 002B  DS 002B
CS 0023  SS 002B

1: [esp+4] 76F96A78 ntdll.76F96A78
2: [esp+8] 76F969FC ntdll.76F969FC
3: [esp+C] 00000000 00000000
4: [esp+10] 00000001 00000001
5: [esp+14] 0019FA18 0019FA18
```

```
Address   Hex                                               ASCII
76F91000  16 00 18 00 F0 7D F9 76 14 00 16 00 50 7C F9 76  ....ð}ùv....P|ùv
76F91010  00 00 02 00 0C 5E F9 76 0E 00 10 00 C8 7F F9 76  .....^ùv....È.ùv
76F91020  0C 00 0E 00 B8 7F F9 76 08 00 0A 00 88 7B F9 76  .......ùv.....{ùv
76F91030  06 00 08 00 98 7F F9 76 06 00 08 00 A8 7F F9 76  ......ùv.... .ùv
76F91040  06 00 08 00 A0 7F F9 76 06 00 08 00 B0 7F F9 76  ....  .ùv....'.ùv
76F91050  1C 00 1E 00 84 7C F9 76 20 00 22 00 40 82 F9 76  .....|ùv .".@.ùv
76F91060  84 00 86 00 B8 81 F9 76 50 6C FC 76 D0 4A 09 77  ......ùvPlüvÐJ.w
76F91070  80 B4 FB 76 10 49 09 77 50 20 FC 76 E0 69 FC 76  . ûv.I.wP üvàiüv
76F91080  90 49 09 77 50 4A 09 77 C0 57 FC 76 D0 4A 09 77  .I.wPJ.wÀWüvÐJ.w
76F91090  E0 25 FC 76 E0 69 FC 76 E0 49 09 77 50 4A 09 77  à%üvàiüvàI.wPJ.w
76F910A0  F0 CF FF 76 D0 4A 09 77 00 00 00 00 00 00 00 00  àÏÿvÐJ.w........
```

```
0019FA18  70E84B20
0019FA1C  76F96A78  ntdll."LdrpInitializeProcess"
0019FA20  76F969FC  ntdll."minkernel\\ntdll\\ldrinit.c"
0019FA24  00000000
0019FA28  00000001
0019FA2C  0019FA18
0019FA30  7703C420  return to ntdll.RtlCaptureStackConte
0019FA34  0019FC9C  Pointer to SEH_Record[1]
0019FA38  7700AE60  ntdll.wcstombs+70
0019FA3C  07F86044
0019FA40  00000000
0019FA44  0019FCAC
0019FA48  7703C431  return to ntdll.RtlCaptureStackConte
0019FA4C  70E84DC8
```

```
Command: Commands are comma separated (like assembly instructions): mov eax, ebx
Paused  System breakpoint reached!                    Time Wasted Debugging: 0:02:05:39
```

Right-click, scroll to "Search for" => "All User Modules" => "String References" to view the mappings of meaningful strings to hexadecimal values stored at addresses in the assembly code.



CrackMe1.exe - PID: 2944 - Module: ntdll.dll - Thread: Main Thread 3904 - x32dbg [Elevated]

```
Address  Disassembly                               String A String
0040243B push crackme1.40D24C                      0040D24C "FlsFree"
00402445 push crackme1.40D24C                      0040D24C "FlsFree"
00402476 push crackme1.40D25C                      0040D25C "FlsSetValue"
00402480 push crackme1.40D25C                      0040D25C "FlsSetValue"
004024B4 push crackme1.40D270                      0040D270 "InitializeCriticalSectionEx"
004024BE push crackme1.40D270                      0040D270 "InitializeCriticalSectionEx"
00403C2E push crackme1.40DACC                      0040DACC L"mscoree.dll"
00403C40 push crackme1.40DAE4                      0040DAE4 "CorExitProcess"
004044B2 mov eax,dword ptr ds:[41363C]             0041363C "`OA"
004058F2 mov eax,dword ptr ds:[40DCEC]             0040DCEC &L"zh-TW"
004058F9 mov eax,dword ptr ds:[40DCE8]             0040DCE8 &L"ko-KR"
00405900 mov eax,dword ptr ds:[40DCE4]             0040DCE4 &L"zh-CN"
00405907 mov eax,dword ptr ds:[40DCE0]             0040DCE0 &L"ja-JP"
00405C7A mov dword ptr ds:[41363C],eax             0041363C "`OA"
004067B7 cmp eax,dword ptr ds:[413654]             00413654 "XBA"
004067C9 cmp eax,dword ptr ds:[413658]             00413658 "XBA"
004067DB cmp eax,dword ptr ds:[41365C]             0041365C "XBA"
004067ED cmp eax,dword ptr ds:[413660]             00413660 "XBA"
004067FF cmp eax,dword ptr ds:[413664]             00413664 "XBA"
00406811 cmp eax,dword ptr ds:[413668]             00413668 "XBA"
00406823 cmp eax,dword ptr ds:[41366C]             0041366C "XBA"
00406835 cmp eax,dword ptr ds:[413680]             00413680 "\\BA"
00406847 cmp eax,dword ptr ds:[413684]             00413684 "\\BA"
00406859 cmp eax,dword ptr ds:[413688]             00413688 "\\BA"
0040686B cmp eax,dword ptr ds:[41368C]             0041368C "\\BA"
0040687D cmp eax,dword ptr ds:[413690]             00413690 "\\BA"
0040688F cmp eax,dword ptr ds:[413694]             00413694 "\\BA"
004068C2 cmp eax,dword ptr ds:[41364C]             0041364C "XBA"
004068D4 cmp eax,dword ptr ds:[413650]             00413650 "XBA"
004068F8 cmp eax,dword ptr ds:[41367C]             0041367C "\\BA"
00406D92 cmp ecx,crackme1.40DD20                   0040DD20 &"Sun"
00406DBC cmp esi,crackme1.40DD20                   0040DD20 &"Sun"
00406DEC cmp ecx,crackme1.40DD20                   0040DD20 &"Sun"
00406F51 push crackme1.40ED0C                      0040ED0C "AreFileApisANSI"
00406F5B push crackme1.40ED0C                      0040ED0C "AreFileApisANSI"
00406F6B push crackme1.40ED94                      0040ED94 "LCMapStringEx"
00406F75 push crackme1.40ED94                      0040ED94 "LCMapStringEx"
00406F85 push crackme1.40EDAC                      0040EDAC "LocaleNameToLCID"
00406F8F push crackme1.40EDAC                      0040EDAC "LocaleNameToLCID"
00406FCE mov ebx,dword ptr ds:[ebx*4+40E7E8]       0040E7E8 &L"api-ms-win-core-datetime-l1-1-1"
00406FF6 push crackme1.40ECE8                      0040ECE8 L"api-ms-"
0040700A push crackme1.40ECF8                      0040ECF8 L"ext-ms-"
004070F1 push crackme1.40EDC4                      0040EDC4 "AppPolicyGetProcessTerminationMethod"
004070FB push crackme1.40EDC4                      0040EDC4 "AppPolicyGetProcessTerminationMethod"
00407150 push crackme1.40ED24                      0040ED24 "FlsAlloc"
0040715A push crackme1.40ED24                      0040ED24 "FlsAlloc"
```

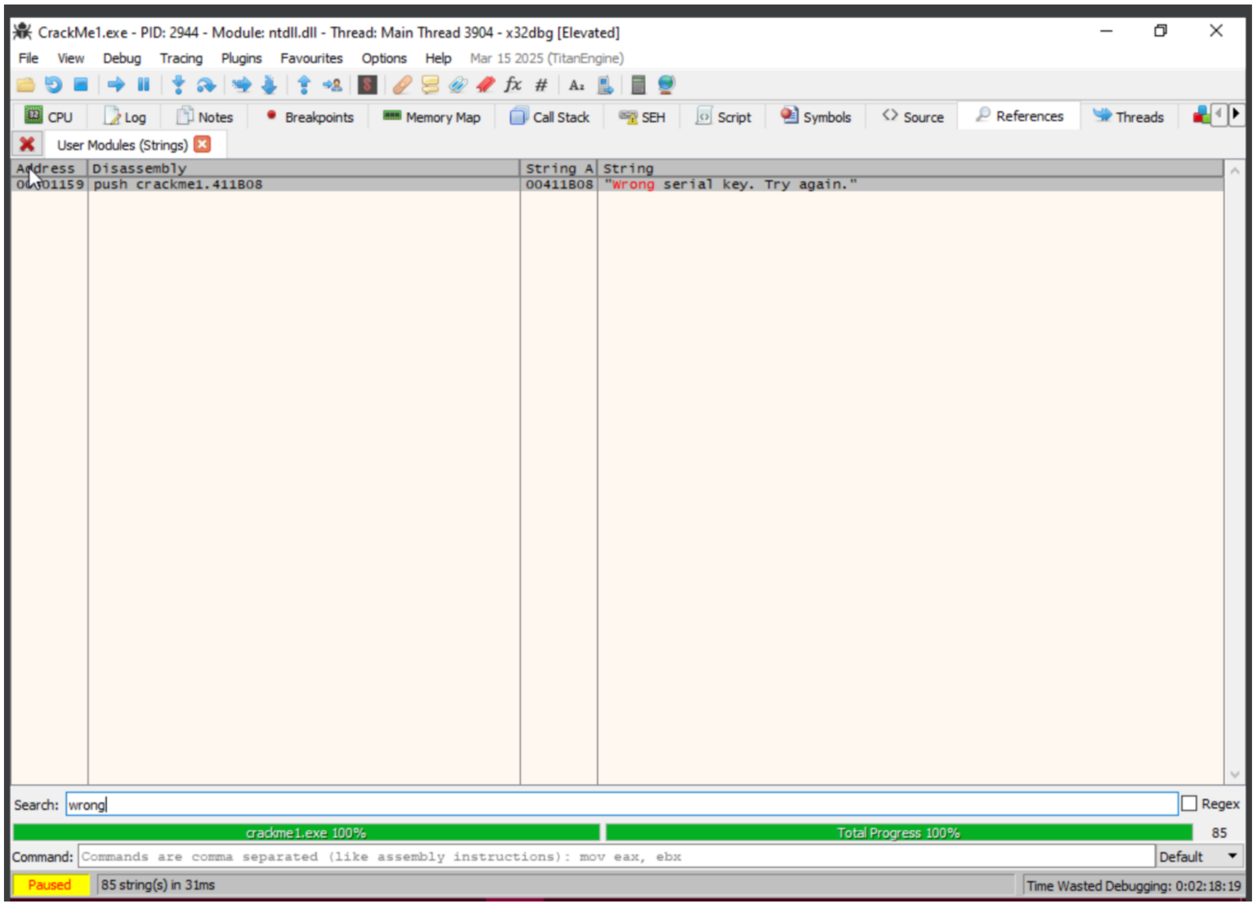
Search: Type here to filter results...

crackme1.exe 100% Total Progress 100% 85

Command: Commands are comma separated (like assembly instructions): mov eax, ebx Default

85 string(s) in 46ms Time Wasted Debugging: 0:02:12:56

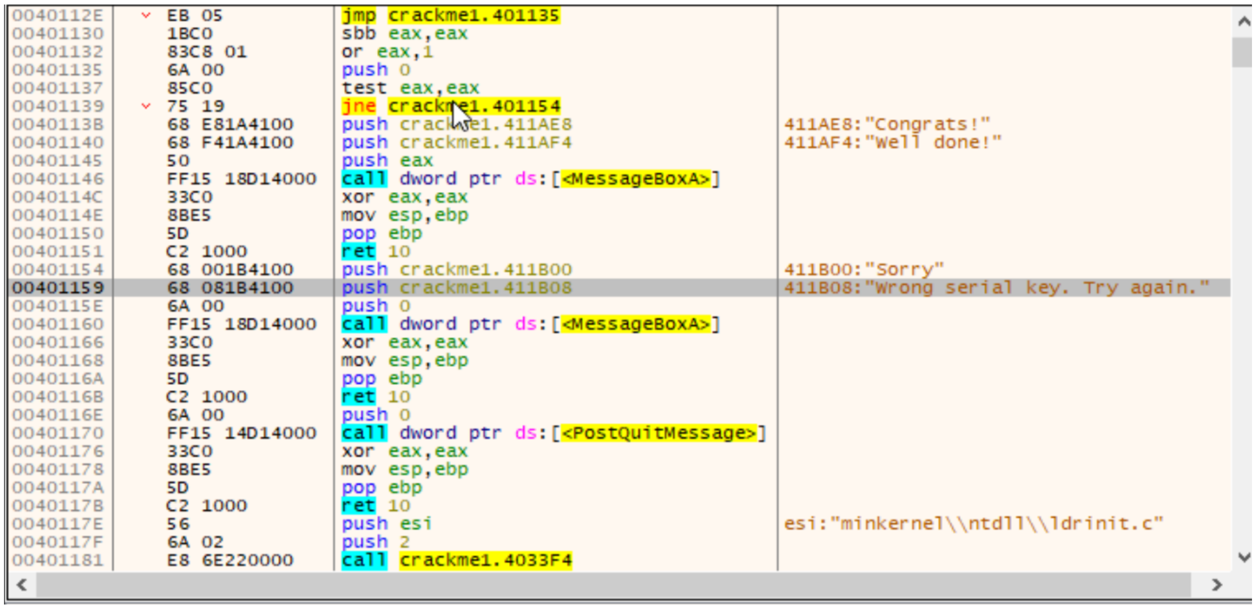
Then search for the string "Wrong serial key. Try again." which we noted during the initial test run of the .exe file.


CrackMe1.exe - PID: 2944 - Module: ntdll.dll - Thread: Main Thread 3904 - x32dbg [Elevated]
File View Debug Tracing Plugins Favourites Options Help Mar 15 2025 (TitanEngine)
CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads
User Modules (Strings)

```
Address  Disassembly              String A String
00401159 push crackme1.411B08     00411B08 "Wrong serial key. Try again."
```

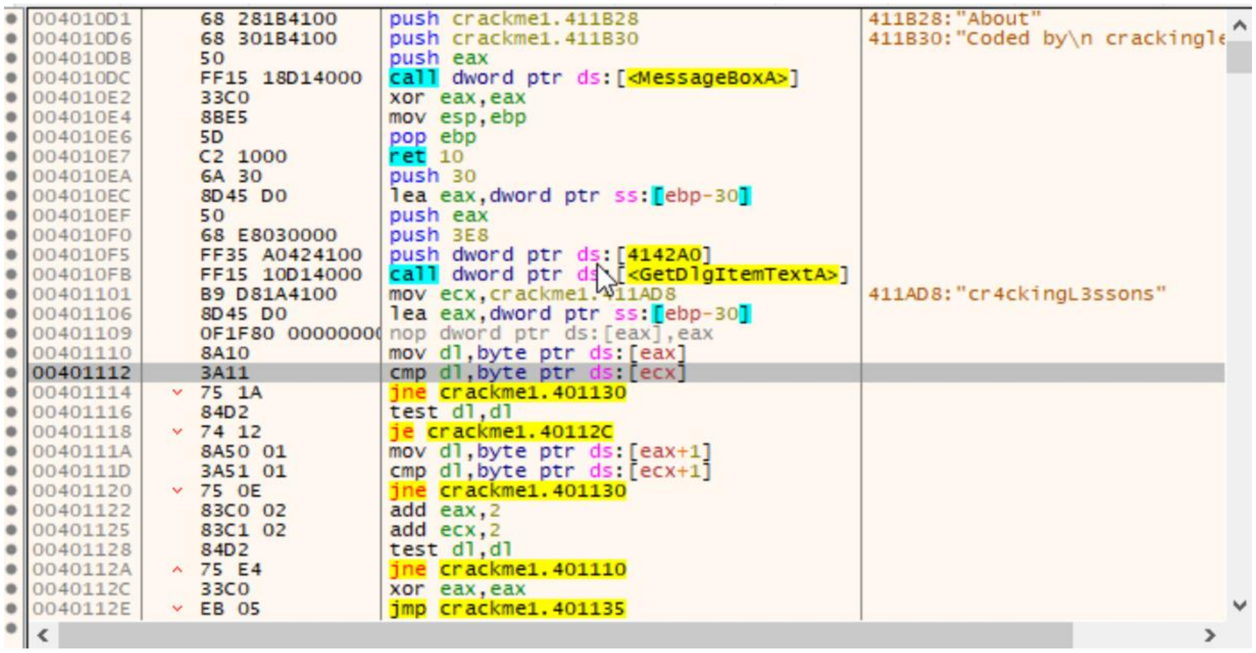
```
Search: wrong
crackme1.exe 100%        Total Progress 100%        85
Command: Commands are comma separated (like assembly instructions): mov eax, ebx    Default
Paused   85 string(s) in 31ms        Time Wasted Debugging: 0:02:18:19
```

Double-click the string "Wrong serial key. Try again" to jump to its location in the assembly code (address 00401159).



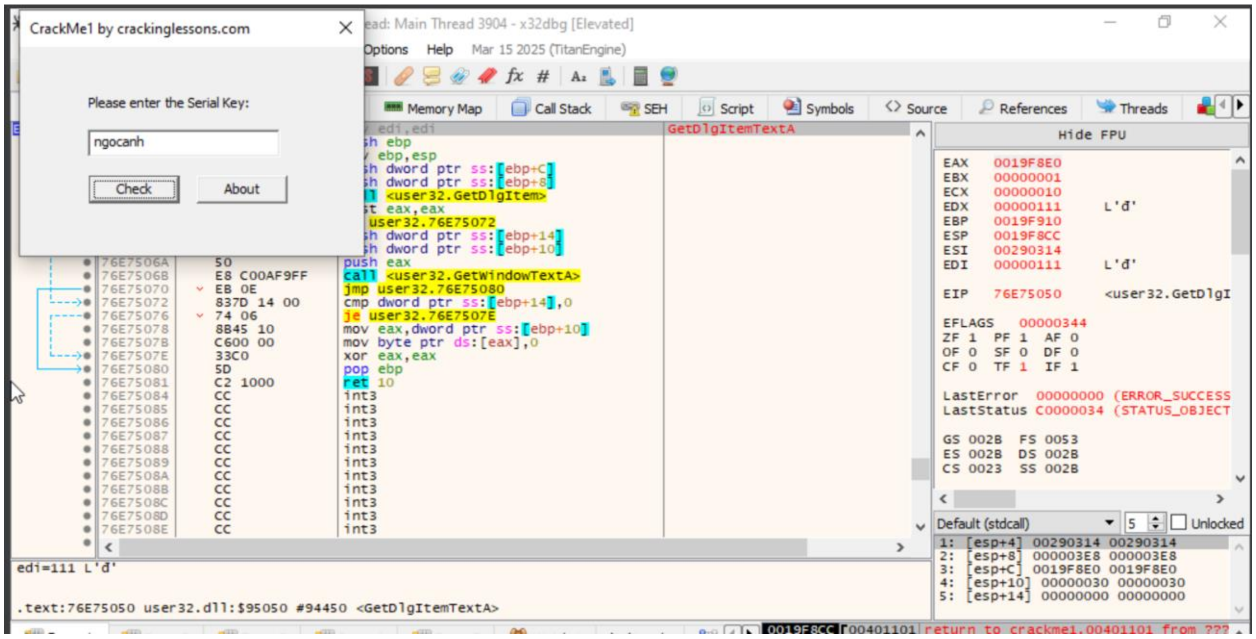
```
0040112E   EB 05         jmp crackme1.401135
00401130   1BC0          sbb eax,eax
00401132   83C8 01       or eax,1
00401135   6A 00         push 0
00401137   85C0          test eax,eax
00401139   75 19         jne crackme1.401154
0040113B   68 E81A4100   push crackme1.411AE8                 411AE8:"Congrats!"
00401140   68 F41A4100   push crackme1.411AF4                 411AF4:"Well done!"
00401145   50            push eax
00401146   FF15 18D14000 call dword ptr ds:[<MessageBoxA>]
0040114C   33C0          xor eax,eax
0040114E   8BE5          mov esp,ebp
00401150   5D            pop ebp
00401151   C2 1000       ret 10
00401154   68 001B4100   push crackme1.411B00                 411B00:"Sorry"
00401159   68 081B4100   push crackme1.411B08                 411B08:"Wrong serial key. Try again."
0040115E   6A 00         push 0
00401160   FF15 18D14000 call dword ptr ds:[<MessageBoxA>]
00401166   33C0          xor eax,eax
00401168   8BE5          mov esp,ebp
0040116A   5D            pop ebp
0040116B   C2 1000       ret 10
0040116E   6A 00         push 0
00401170   FF15 14D14000 call dword ptr ds:[<PostQuitMessage>]
00401176   33C0          xor eax,eax
00401178   8BE5          mov esp,ebp
0040117A   5D            pop ebp
0040117B   C2 1000       ret 10
0040117E   56            push esi                             esi:"minkernel\\ntdll\\ldrinit.c"
0040117F   6A 02         push 2
00401181   E8 6E220000   call crackme1.4033F4
```

Note: This string is the message shown when the serial key is incorrect. Locating this string helps identify the code section that checks the serial key, typically just before the string is called.
From the "Wrong serial key. Try again" string at address 00401159, scroll up to find the CMP instruction that performs the serial key comparison. The CMP instruction is found at 00401112, where "MOV ECX, crackme1.411AD8" assigns the value "cr4ckingL3ssons" to ECX. Then, CMP compares DL (each byte of input string) with ECX. If they don't match, the program jumps to the failure branch and displays "Wrong serial key. Try again".



```
004010D1    68 281B4100      push crackme1.411B28                  411B28:"About"
004010D6    68 301B4100      push crackme1.411B30                  411B30:"Coded by\n crackingle
004010DB    50               push eax
004010DC    FF15 18D14000    call dword ptr ds:[<MessageBoxA>]
004010E2    33C0             xor eax,eax
004010E4    8BE5             mov esp,ebp
004010E6    5D               pop ebp
004010E7    C2 1000          ret 10
004010EA    6A 30            push 30
004010EC    8D45 D0          lea eax,dword ptr ss:[ebp-30]
004010EF    50               push eax
004010F0    68 E8030000      push 3E8
004010F5    FF35 A0424100    push dword ptr ds:[4142A0]
004010FB    FF15 10D14000    call dword ptr ds:[<GetDlgItemTextA>]
00401101    B9 D81A4100      mov ecx,crackme1.411AD8               411AD8:"cr4ckingL3ssons"
00401106    8D45 D0          lea eax,dword ptr ss:[ebp-30]
00401109    0F1F80 0000000   nop dword ptr ds:[eax],eax
00401110    8A10             mov dl,byte ptr ds:[eax]
00401112    3A11             cmp dl,byte ptr ds:[ecx]
00401114  ˅ 75 1A            jne crackme1.401130
00401116    84D2             test dl,dl
00401118  ˅ 74 12            je crackme1.40112C
0040111A    8A50 01          mov dl,byte ptr ds:[eax+1]
0040111D    3A51 01          cmp dl,byte ptr ds:[ecx+1]
00401120  ˅ 75 0E            jne crackme1.401130
00401122    83C0 02          add eax,2
00401125    83C1 02          add ecx,2
00401128    84D2             test dl,dl
0040112A  ˄ 75 E4            jne crackme1.401110
0040112C    33C0             xor eax,eax
0040112E  ˅ EB 05            jmp crackme1.401135
```

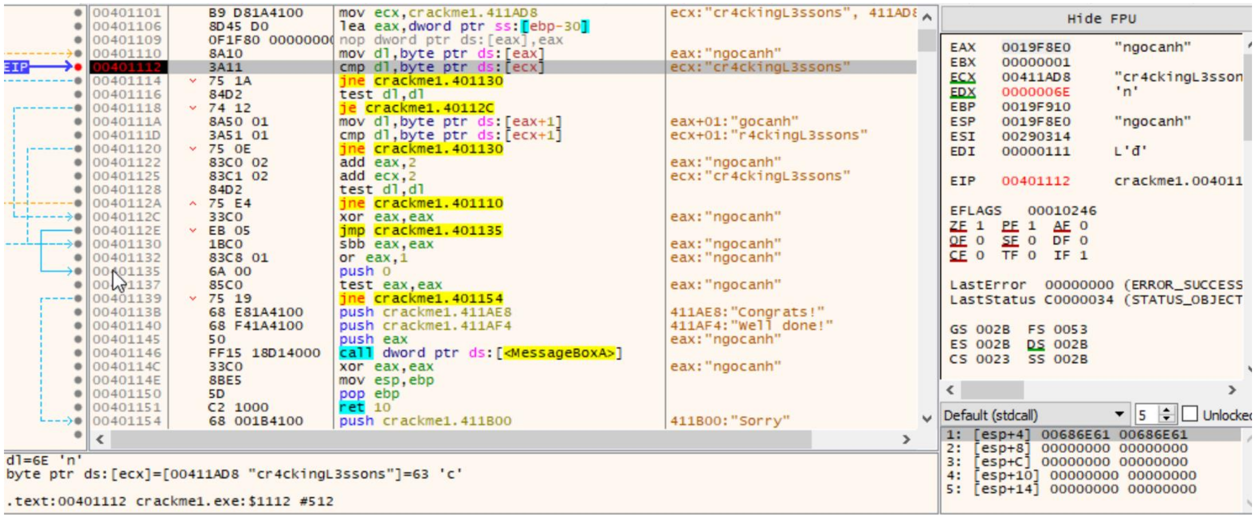
Set a breakpoint at address 00401112 to inspect the values of registers (such as DL and ECX) during the serial key comparison, aiming to identify the correct value. Press F9 to execute the program, enter a random key ("ngocanh"), and the program will pause at the breakpoint for analysis.


CrackMe1 by crackinglessons.com
Please enter the Serial Key:
ngocanh
Check
About
ead: Main Thread 3904 - x32dbg [Elevated]
Options
Help
Mar 15 2025 (TitanEngine)
Memory Map
Call Stack
SEH
Script
Symbols
Source
References
Threads
Hide FPU
Default (stdcall)
5
Unlocked

```
                              / edi,edi                               GetDlgItemTextA
                              sh ebp
                              / ebp,esp
                              sh dword ptr ss:[ebp+C]
                              sh dword ptr ss:[ebp+8]
                              ll <user32.GetDlgItem>
                              st eax,eax
                               user32.76E75072
                              sh dword ptr ss:[ebp+14]
                              sh dword ptr ss:[ebp+10]
76E7506A       50             push eax
76E7506B       E8 C00AF9FF    call <user32.GetWindowTextA>
76E75070       EB 0E          jmp user32.76E75080
76E75072       837D 14 00     cmp dword ptr ss:[ebp+14],0
76E75076       74 06          je user32.76E7507E
76E75078       8B45 10        mov eax,dword ptr ss:[ebp+10]
76E7507B       C600 00        mov byte ptr ds:[eax],0
76E7507E       33C0           xor eax,eax
76E75080       5D             pop ebp
76E75081       C2 1000        ret 10
76E75084       CC             int3
76E75085       CC             int3
76E75086       CC             int3
76E75087       CC             int3
76E75088       CC             int3
76E75089       CC             int3
76E7508A       CC             int3
76E7508B       CC             int3
76E7508C       CC             int3
76E7508D       CC             int3
76E7508E       CC             int3

edi=111 L'đ'

.text:76E75050 user32.dll:$95050 #94450 <GetDlgItemTextA>
```

```
EAX    0019F8E0
EBX    00000001
ECX    00000010
EDX    00000111     L'đ'
EBP    0019F910
ESP    0019F8CC
ESI    00290314
EDI    00000111     L'đ'

EIP    76E75050     <user32.GetDlgI

EFLAGS    00000344
ZF 1  PF 1  AF 0
OF 0  SF 0  DF 0
CF 0  TF 1  IF 1

LastError  00000000 (ERROR_SUCCESS
LastStatus C0000034 (STATUS_OBJECT

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

1: [esp+4] 00290314 00290314
2: [esp+8] 000003E8 000003E8
3: [esp+C] 0019F8E0 0019F8E0
4: [esp+10] 00000030 00000030
5: [esp+14] 00000000 00000000

0019F8CC [00401101 return to crackme1.00401101 from ???
```

Press F8 to step through the code until the breakpoint, focusing on the register window with variables EAX, ECX, and ESP. Observe their changes after each code step.

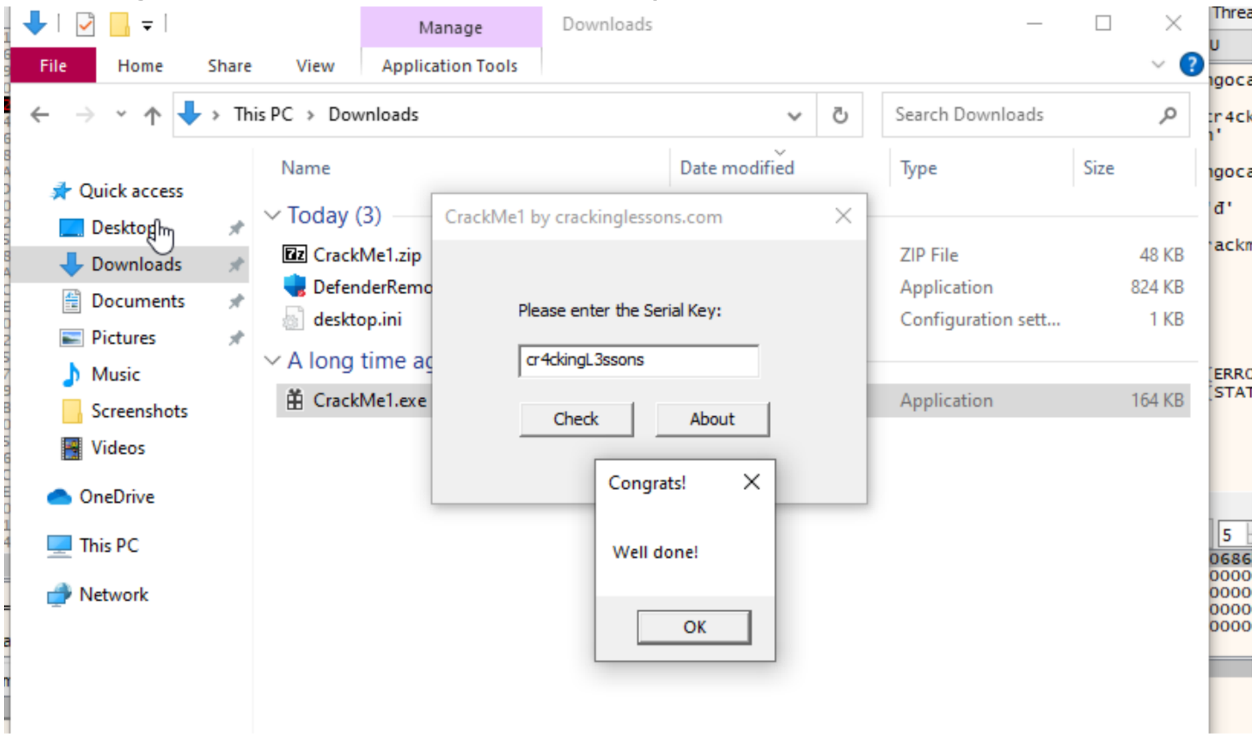


```
00401101   B9 D81A4100        mov ecx,crackme1.411AD8            ecx:"cr4ckingL3ssons", 411AD8
00401106   8D45 D0            lea eax,dword ptr ss:[ebp-30]
00401109   0F1F80 00000000    nop dword ptr ds:[eax],eax
00401110   8A10               mov dl,byte ptr ds:[eax]           eax:"ngocanh"
00401112   3A11               cmp dl,byte ptr ds:[ecx]           ecx:"cr4ckingL3ssons"
00401114   75 1A              jne crackme1.401130
00401116   84D2               test dl,dl
00401118   74 12              je crackme1.40112C
0040111A   8A50 01            mov dl,byte ptr ds:[eax+1]         eax+01:"gocanh"
0040111D   3A51 01            cmp dl,byte ptr ds:[ecx+1]         ecx+01:"r4ckingL3ssons"
00401120   75 0E              jne crackme1.401130
00401122   83C0 02            add eax,2                          eax:"ngocanh"
00401125   83C1 02            add ecx,2                          ecx:"cr4ckingL3ssons"
00401128   84D2               test dl,dl
0040112A   75 E4              jne crackme1.401110
0040112C   33C0               xor eax,eax                        eax:"ngocanh"
0040112E   EB 05              jmp crackme1.401135
00401130   1BC0               sbb eax,eax                        eax:"ngocanh"
00401132   83C8 01            or eax,1                           eax:"ngocanh"
00401135   6A 00              push 0
00401137   85C0               test eax,eax                       eax:"ngocanh"
00401139   75 19              jne crackme1.401154
0040113B   68 E81A4100        push crackme1.411AE8               411AE8:"Congrats!"
00401140   68 F41A4100        push crackme1.411AF4               411AF4:"Well done!"
00401145   50                 push eax                           eax:"ngocanh"
00401146   FF15 18D14000      call dword ptr ds:[<MessageBoxA>]
0040114C   33C0               xor eax,eax                        eax:"ngocanh"
0040114E   8BE5               mov esp,ebp
00401150   5D                 pop ebp
00401151   C2 1000            ret 10
00401154   68 001B4100        push crackme1.411B00               411B00:"Sorry"

dl=6E 'n'
byte ptr ds:[ecx]=[00411AD8 "cr4ckingL3ssons"]=63 'c'

.text:00401112 crackme1.exe:$1112 #512
```

```
EAX    0019F8E0     "ngocanh"
EBX    00000001
ECX    00411AD8     "cr4ckingL3sson
EDX    0000006E     'n'
EBP    0019F910
ESP    0019F8E0     "ngocanh"
ESI    00290314
EDI    00000111     L'đ'

EIP    00401112     crackme1.004011

EFLAGS    00010246
ZF 1  PF 1  AF 0
OF 0  SF 0  DF 0
CF 0  TF 0  IF 1

LastError  00000000 (ERROR_SUCCESS
LastStatus C0000034 (STATUS_OBJECT

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

1: [esp+4] 00686E61 00686E61
2: [esp+8] 00000000 00000000
3: [esp+C] 00000000 00000000
4: [esp+10] 00000000 00000000
5: [esp+14] 00000000 00000000
```

At the breakpoint, we observe that the first byte of EAX (stored in DL) is 'n', while the ECX string starts with 'c'. This raises doubts about the accuracy of

"cr4ckingL3ssons" as the correct serial key.



Name	Type	Size
CrackMe1.zip	ZIP File	48 KB
DefenderRemo	Application	824 KB
desktop.ini	Configuration sett...	1 KB
CrackMe1.exe	Application	164 KB

CrackMe1 by crackinglessons.com

Please enter the Serial Key:

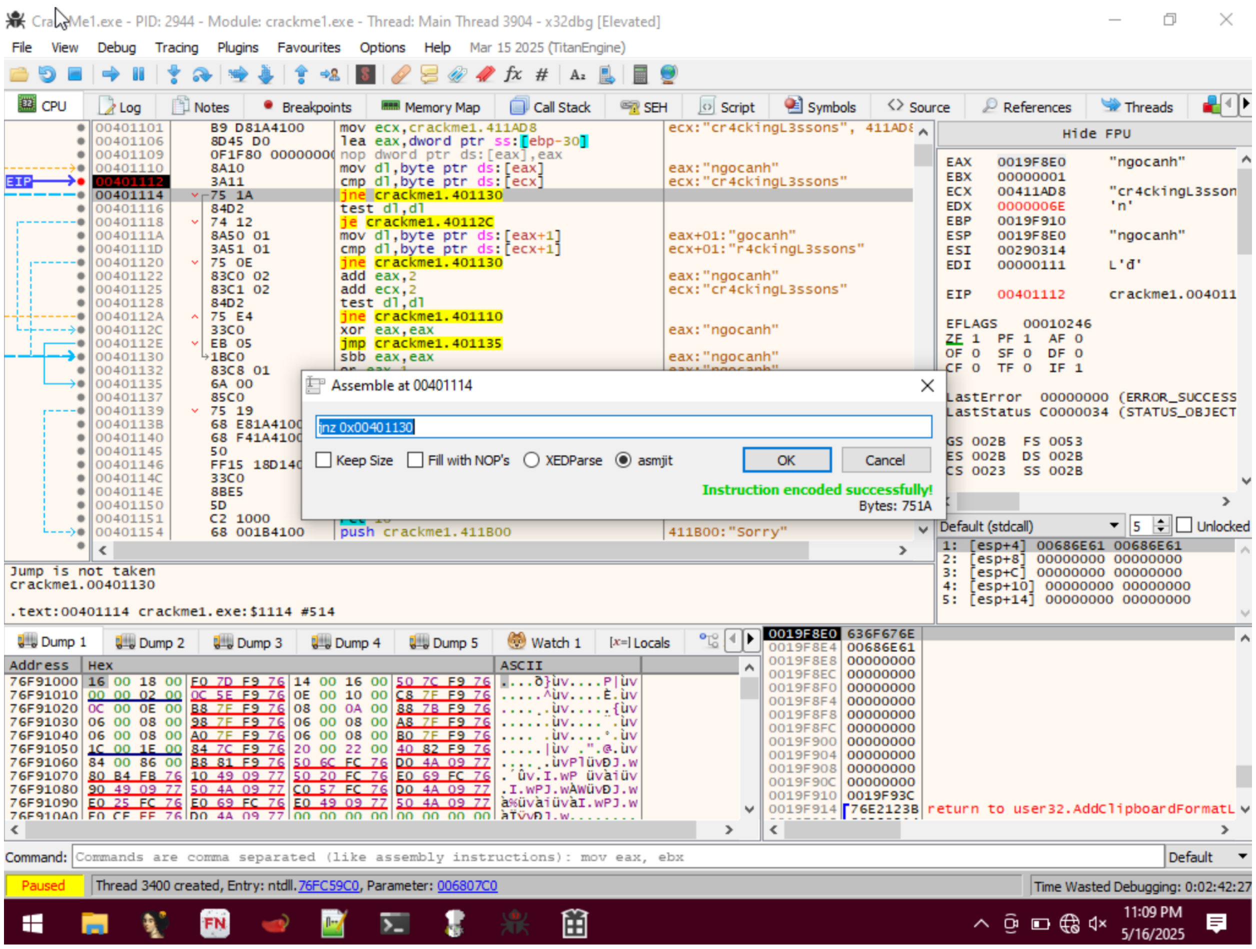
cr4ckingL3ssons

Congrats!

Well done!

Test the CrackMe1.exe file, resulting in the message "Well done!"

Conclusion: The serial key is "cr4ckingL3ssons".


CrackMe1.exe - PID: 2944 - Module: crackme1.exe - Thread: Main Thread 3904 - x32dbg [Elevated]
File View Debug Tracing Plugins Favourites Options Help Mar 15 2025 (TitanEngine)
CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads

```
00401101   B9 D81A4100       mov ecx,crackme1.411AD8          ecx:"cr4ckingL3ssons", 411AD8
00401106   8D45 D0           lea eax,dword ptr ss:[ebp-30]
00401109   0F1F80 0000000    nop dword ptr ds:[eax],eax
00401110   8A10              mov dl,byte ptr ds:[eax]         eax:"ngocanh"
00401112   3A11              cmp dl,byte ptr ds:[ecx]         ecx:"cr4ckingL3ssons"
00401114   75 1A             jne crackme1.401130
00401116   84D2              test dl,dl
00401118   74 12             je crackme1.40112C
0040111A   8A50 01           mov dl,byte ptr ds:[eax+1]       eax+01:"gocanh"
0040111D   3A51 01           cmp dl,byte ptr ds:[ecx+1]       ecx+01:"r4ckingL3ssons"
00401120   75 0E             jne crackme1.401130
00401122   83C0 02           add eax,2                        eax:"ngocanh"
00401125   83C1 02           add ecx,2                        ecx:"cr4ckingL3ssons"
00401128   84D2              test dl,dl
0040112A   75 E4             jne crackme1.401110
0040112C   33C0              xor eax,eax                      eax:"ngocanh"
0040112E   EB 05             jmp crackme1.401135
00401130   1BC0              sbb eax,eax                      eax:"ngocanh"
00401132   83C8 01
00401135   6A 00
00401137   85C0
00401139   75 19
0040113B   68 E81A4100
00401140   68 F41A4100
00401145   50
00401146   FF15 18D140
0040114C   33C0
0040114E   8BE5
00401150   5D
00401151   C2 1000
00401154   68 001B4100       push crackme1.411B00             411B00:"Sorry"
```

Assemble at 00401114

`jnz 0x00401130`

[ ] Keep Size [ ] Fill with NOP's ( ) XEDParse [✓ asmjit] OK Cancel

Instruction encoded successfully!
Bytes: 751A

```
Jump is not taken
crackme1.00401130

.text:00401114 crackme1.exe:$1114 #514
```

Hide FPU

```
EAX    0019F8E0     "ngocanh"
EBX    00000001
ECX    00411AD8     "cr4ckingL3sson
EDX    0000006E     'n'
EBP    0019F910
ESP    0019F8E0     "ngocanh"
ESI    00290314
EDI    00000111     L'đ'

EIP    00401112     crackme1.004011

EFLAGS   00010246
ZF 1  PF 1  AF 0
OF 0  SF 0  DF 0
CF 0  TF 0  IF 1

LastError  00000000 (ERROR_SUCCESS
LastStatus C0000034 (STATUS_OBJECT

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

Default (stdcall) 5 [ ] Unlocked

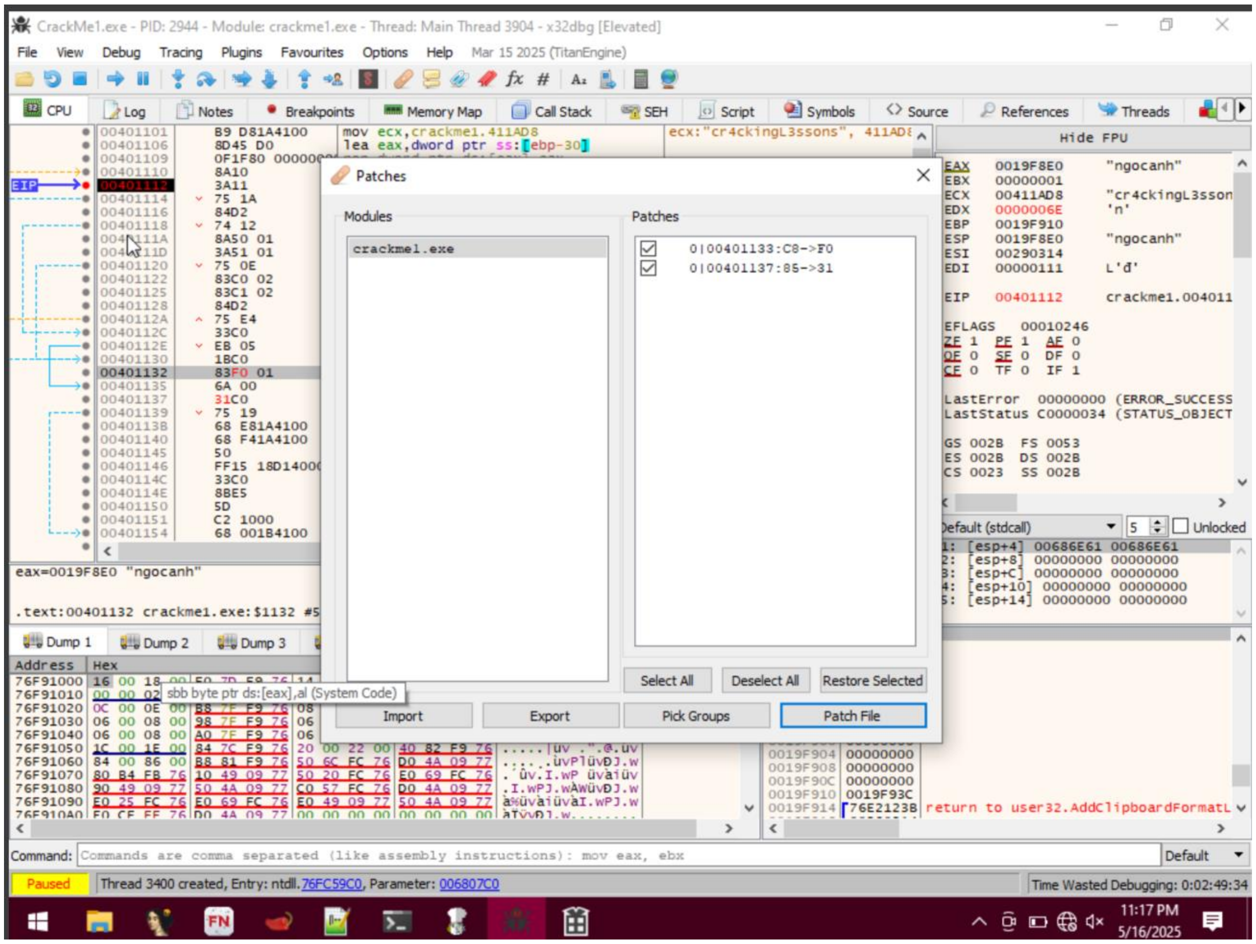
```
1:  [esp+4]  00686E61 00686E61
2:  [esp+8]  00000000 00000000
3:  [esp+C]  00000000 00000000
4:  [esp+10] 00000000 00000000
5:  [esp+14] 00000000 00000000
```

```
Address   Hex                                              ASCII
76F91000  16 00 18 00 F0 7D F9 76 14 00 16 00 50 7C F9 76  ....ð}ùv....P|ùv
76F91010  00 00 02 00 0C 5E F9 76 0E 00 10 00 C8 7F F9 76  .....^ùv....È.ùv
76F91020  0C 00 0E 00 B8 7F F9 76 08 00 0A 00 88 7B F9 76  ......ùv.....{ùv
76F91030  06 00 08 00 98 7F F9 76 06 00 08 00 A8 7F F9 76  ......ùv......ùv
76F91040  06 00 08 00 A0 7F F9 76 06 00 08 00 B0 7F F9 76  ......ùv....°.ùv
76F91050  1C 00 1E 00 84 7C F9 76 20 00 22 00 40 82 F9 76  .....|ùv .".@.ùv
76F91060  84 00 86 00 B8 81 F9 76 50 6C FC 76 D0 4A 09 77  ......ùvPlüvÐJ.w
76F91070  80 B4 FB 76 10 49 09 77 50 20 FC 76 E0 69 FC 76  .´ûv.I.wP üvàiüv
76F91080  90 49 09 77 50 4A 09 77 C0 57 FC 76 D0 4A 09 77  .I.wPJ.wÀWüvÐJ.w
76F91090  E0 25 FC 76 E0 69 FC 76 E0 49 09 77 50 4A 09 77  à%üvàiüvàI.wPJ.w
```

```
0019F8E0  636F676E
0019F8E4  00686E61
0019F8E8  00000000
0019F8EC  00000000
0019F8F0  00000000
0019F8F4  00000000
0019F8F8  00000000
0019F8FC  00000000
0019F900  00000000
0019F904  00000000
0019F908  00000000
0019F90C  00000000
0019F910  0019F93C
0019F914  76E2123B  return to user32.AddClipboardFormatL
```

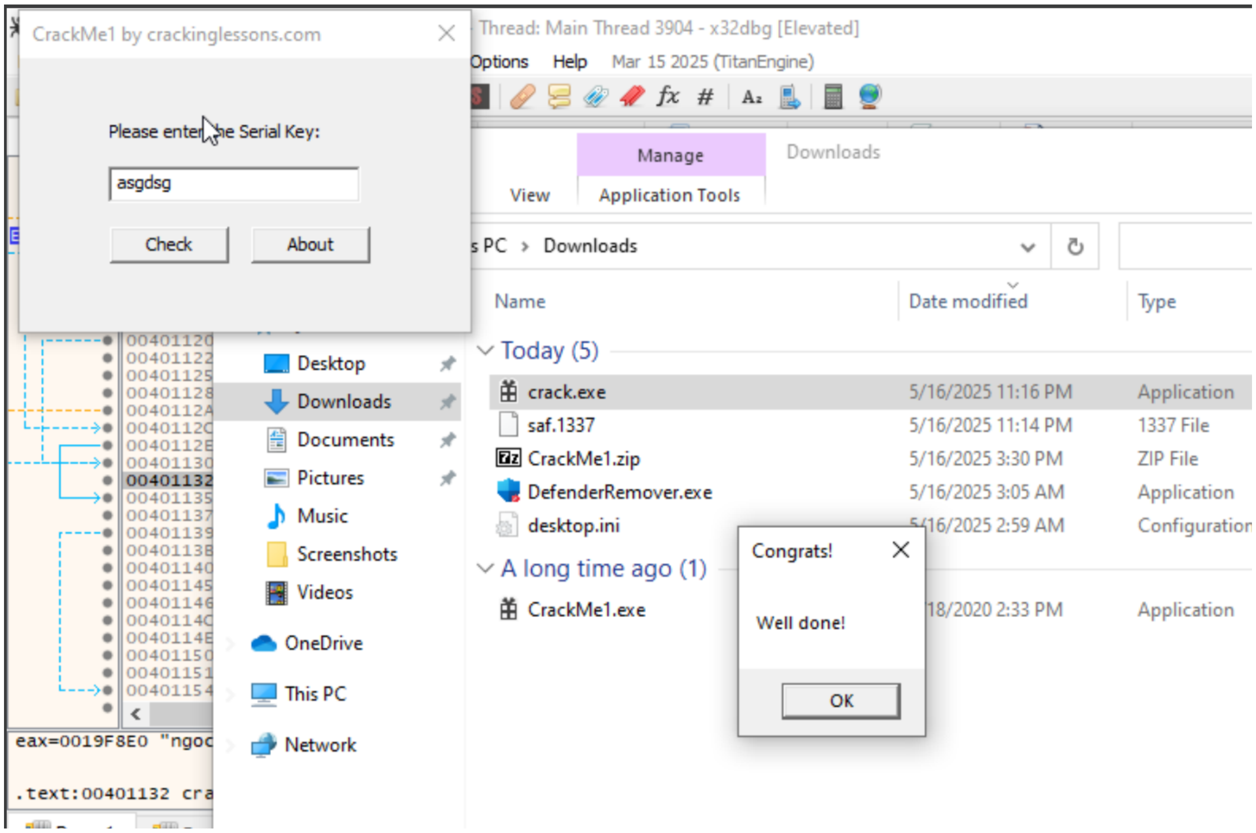
Command: Commands are comma separated (like assembly instructions): mov eax, ebx

Paused   Thread 3400 created, Entry: ntdll.76FC59C0, Parameter: 006807C0   Time Wasted Debugging: 0:02:42:27



```
CrackMe1.exe - PID: 2944 - Module: crackme1.exe - Thread: Main Thread 3904 - x32dbg [Elevated]

00401101  B9 D81A4100      mov ecx,crackme1.411AD8        ecx:"cr4ckingL3ssons", 411AD8
00401106  8D45 D0          lea eax,dword ptr ss:[ebp-30]
00401109  0F1F80 000000
00401110  8A10
00401112  3A11
00401114  75 1A
00401116  84D2
00401118  74 12
0040111A  8A50 01
0040111D  3A51 01
00401120  75 0E
00401122  83C0 02
00401125  83C1 02
00401128  84D2
0040112A  75 E4
0040112C  33C0
0040112E  EB 05
00401130  1BC0
00401132  83F0 01
00401135  6A 00
00401137  31C0
00401139  75 19
0040113B  68 E81A4100
00401140  68 F41A4100
00401145  50
00401146  FF15 18D14000
0040114C  33C0
0040114E  8BE5
00401150  5D
00401151  C2 1000
00401154  68 001B4100

eax=0019F8E0 "ngocanh"

.text:00401132 crackme1.exe:$1132 #5

Patches
Modules                 Patches
crackme1.exe            [✓] 0|00401133:C8->F0
                        [✓] 0|00401137:85->31

Select All   Deselect All   Restore Selected
Import   Export   Pick Groups   Patch File

EAX  0019F8E0   "ngocanh"
EBX  00000001
ECX  00411AD8   "cr4ckingL3sson
EDX  0000006E   'n'
EBP  0019F910
ESP  0019F8E0   "ngocanh"
ESI  00290314
EDI  00000111   L'đ'

EIP  00401112   crackme1.004011

EFLAGS   00010246
ZF 1  PF 1  AF 0
OF 0  SF 0  DF 0
CF 0  TF 0  IF 1

LastError  00000000 (ERROR_SUCCESS
LastStatus C0000034 (STATUS_OBJECT

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

Default (stdcall)   5   Unlocked
1: [esp+4] 00686E61 00686E61
2: [esp+8] 00000000 00000000
3: [esp+C] 00000000 00000000
4: [esp+10] 00000000 00000000
5: [esp+14] 00000000 00000000

Address   Hex
76F91000  16 00 18 00 F0 7D F9 76 14
76F91010  00 00 02
76F91020  0C 00 0E 00 B8 7F F9 76 08
76F91030  06 00 08 00 98 7F F9 76 06
76F91040  06 00 08 00 A0 7F F9 76 06
76F91050  1C 00 1E 00 84 7C F9 76 20 00 22 00 40 82 F9 76  .....|uv .".@.uv
76F91060  84 00 86 00 B8 81 F9 76 50 6C FC 76 D0 4A 09 77  ......uvPlüvÐJ.w
76F91070  80 B4 FB 76 10 49 09 77 50 20 FC 76 E0 69 FC 76  .´ûv.I.wP üvàiüv
76F91080  90 49 09 77 50 4A 09 77 C0 57 FC 76 D0 4A 09 77  .I.wPJ.wÀWüvÐJ.w
76F91090  E0 25 FC 76 E0 69 FC 76 E0 49 09 77 50 4A 09 77  à%üvàiüvàI.wPJ.w

sbb byte ptr ds:[eax],al (System Code)

0019F904  00000000
0019F908  00000000
0019F90C  00000000
0019F910  0019F93C
0019F914  76E2123B  return to user32.AddClipboardFormatL

Command: Commands are comma separated (like assembly instructions): mov eax, ebx
Paused   Thread 3400 created, Entry: ntdll.76FC59C0, Parameter: 006807C0
Time Wasted Debugging: 0:02:49:34
11:17 PM
5/16/2025
```

Convert test eax,eax to xor eax eax and patch file



Name	Date modified	Type
crack.exe	5/16/2025 11:16 PM	Application
saf.1337	5/16/2025 11:14 PM	1337 File
CrackMe1.zip	5/16/2025 3:30 PM	ZIP File
DefenderRemover.exe	5/16/2025 3:05 AM	Application
desktop.ini	5/16/2025 2:59 AM	Configuration
CrackMe1.exe	18/2020 2:33 PM	Application

```
00401120
00401122
00401125
00401128
0040112A
0040112C
0040112E
00401130
00401132
00401135
00401137
00401139
0040113B
00401140
00401145
00401146
0040114C
0040114E
00401150
00401151
00401154
eax=0019F8E0 "ngoc
.text:00401132 cra
```

then check random key and bypass!