Software Cracking Pitch

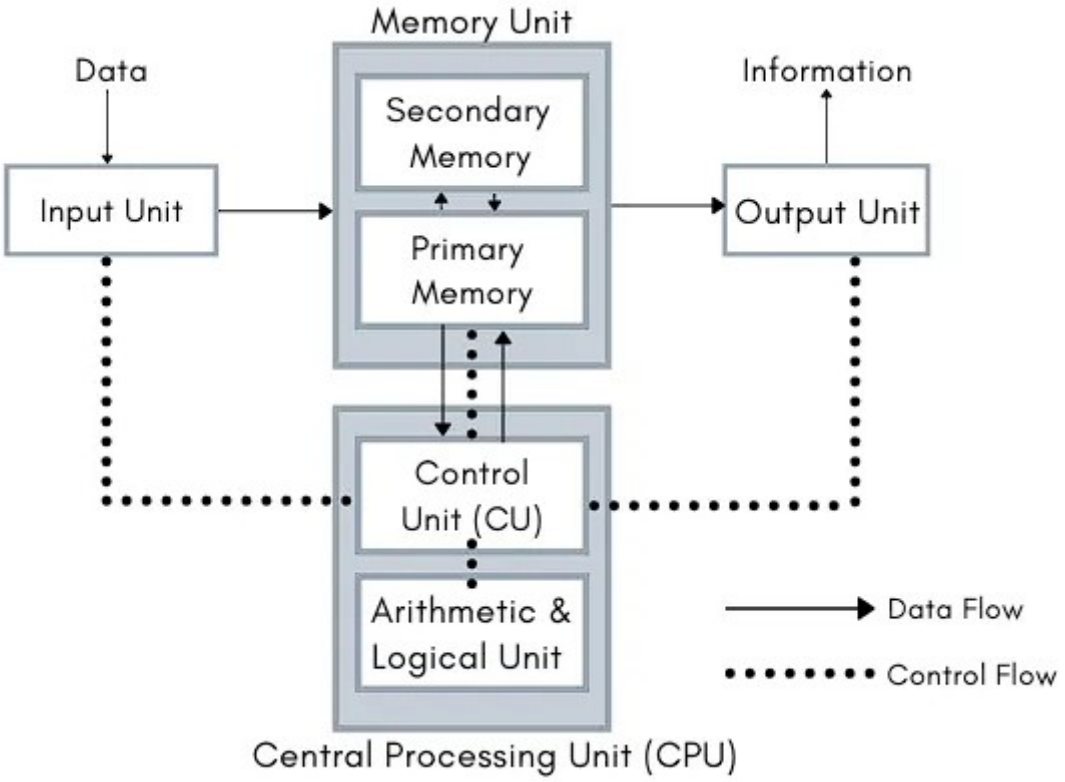
Objective - what is software cracking and why software cracking
Demonstration - How it can be done
Contribution - On people's life
Impact - In long term

Hello Everyone, I am Ankit Raj, an intern at hackveda and today i'll be giving you a demostration on the topic of Software Cracking. So the first and formost objective of this demonstration will be the what is Software cracking or why software cracking is done. Then we will move onto the demonstration part that How this whole process of software cracking is done. And Then we will see the contribution of software cracking on people's life and Finally we will move onto the final part of this demostration which is impacts of using cracked softwares.

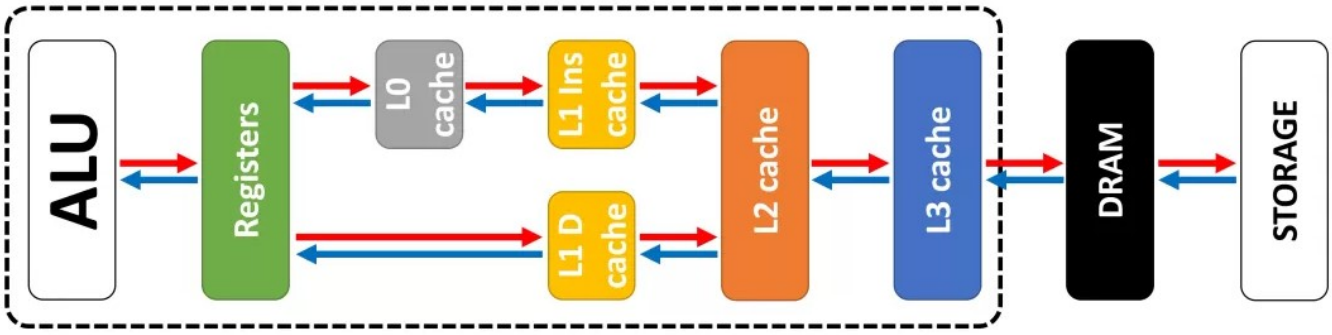
So Let me start you by telling what is software cracking? Software cracking is a process to bypass the limitations of a software and manipulate the software codes in such a way it was not intended to work as or we can simply say Software cracking is the modification of software to remove or disable features especially copy protection features including protection against the manipulation of software, serial number, hardware key.

So, The Question is Why Software Cracking?
Let me tell you at some point of time in your life you may have encountered that you need a software that actually needs to be purchased to be used but you as an end user don't wanna spend a penny to buy it. But you also know there no other softwares better than this. So the point is what you'll do at zero cost? The Answer is Software Cracking. You can either crack the software you wanna use by yourself either you can download the cracked version of that particular software.

Now we will see the demonstration that how it can be done?
First in the demostration we will see the Computer Architecture and working of CPU and Memory Subsytem together :-
So i am gonna give you brief idea of Computer architecture and CPU&Memory subsystem that would be relevent in order to understand process of software cracking.

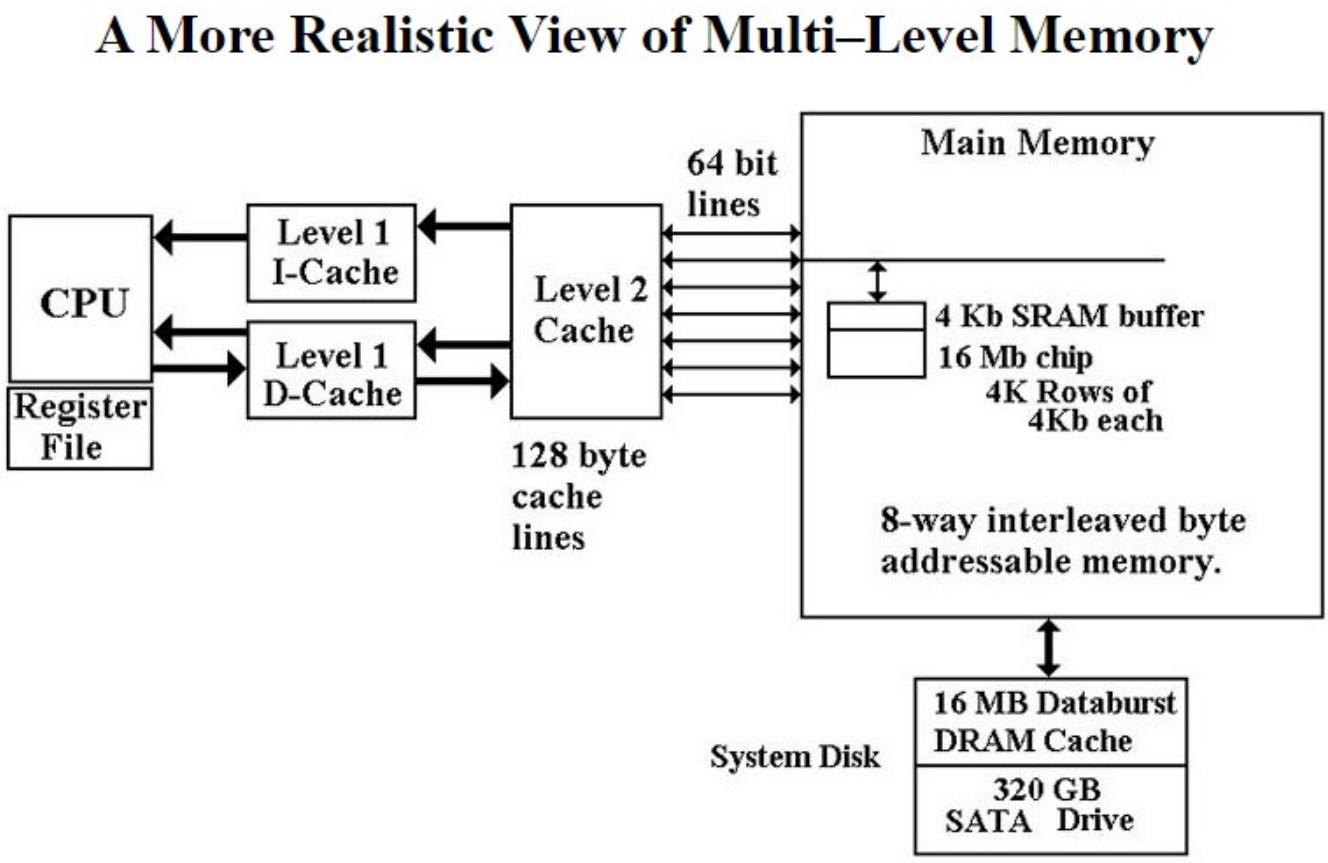


As per the diagram you can see a CPU alongside with Memory Unit, Input Unit and the Output Unit. And the CPU is further divided into some sections that is Control Unit and ALU. As you can see in diagram the control unit controls eveything from ALU to Memory Subsystem and I/O devices. So whenever some program is executed from the secondary memory it gets loaded into the primary memory. After that the CPU fetches the instruction from the Primary memory to execute them. But while execution some intermediatary results are generated that needs to be stored somewhere. That's where registers comes into play!



Now from the above diagram we can the organization of CPU Subsystem along side with Memory Subsystem. CPU subsytem consists of Registers and Cache memories. As told the Registers holds and provide the CPU with intermediate results that gets generated while execution of instruction. But the cache memories holds those results of those instruction that are completely executed.
So we can say registers holds the RAW data while the cache hold the processed data feeds the CPU if needed for faster access.

Below is the diagram you can refer for a more realistic view of multi level memory.



There are various types of registers each with different purpose of work such as GPP, Accumulator register which sits between AU and LU and holds their intermediate results. While some other registers such as Program Counter, Instruction Register have other different purposes. Most relevent registers in scope with this demostration are Stack Pointer register and Base Pointer register. Let me briefly tell you that whatever program is loaded in the CPU for execution the Stack pointer register will hold the Highest most address of the loaded program that means the stack pointer register points to whatever address is at the top of the stack of loaded program while the base pointer register points to the previous frame's base pointer. Usually base pointer is set to stack pointer at the start of the function.

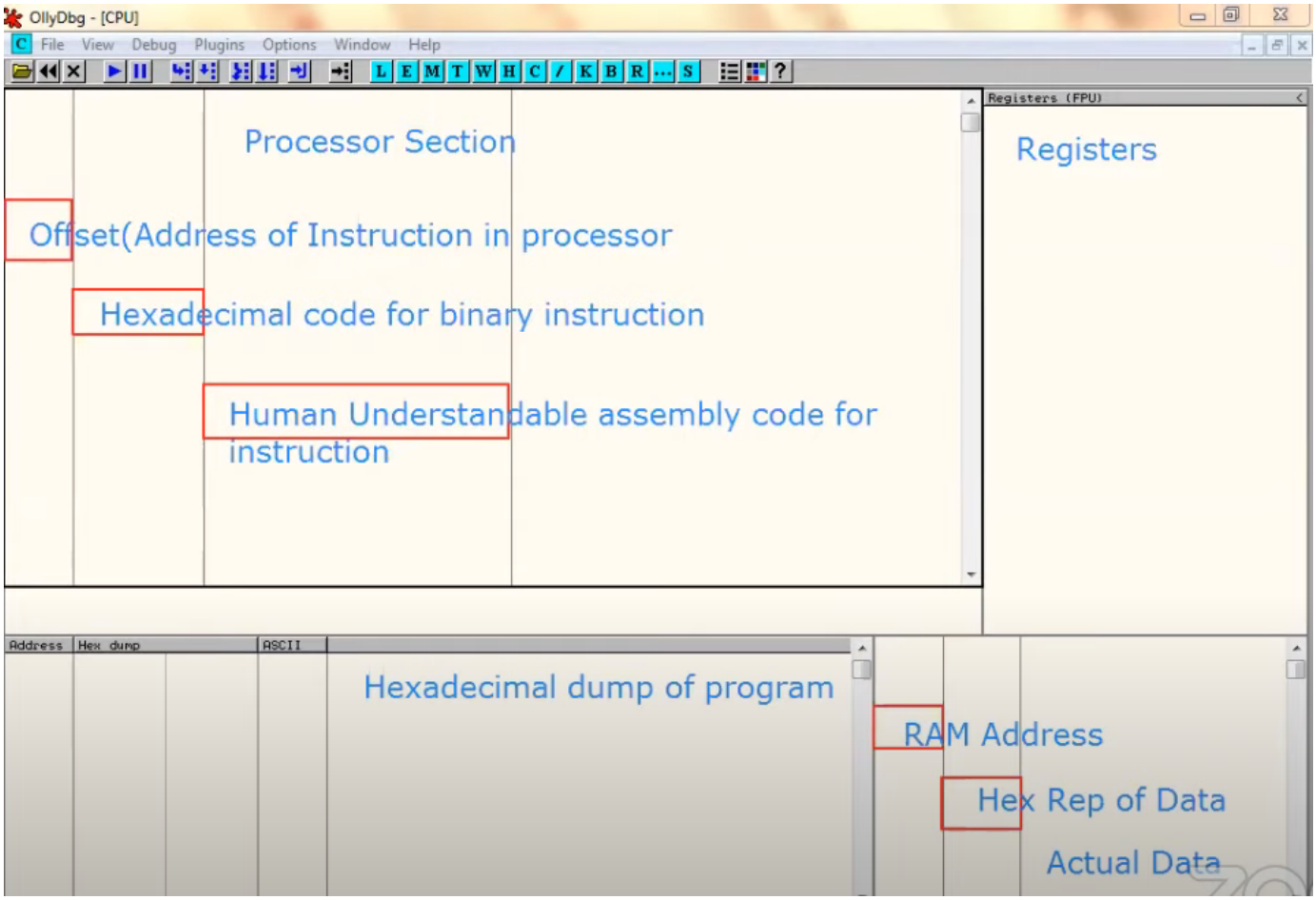
So moving onto the tools and requirement section we need:-

1. A Software which needs to cracked in our case it is open source crackme software.
2. A Disassembler tool such as Olly Debugger which is a 32 bit disassembler.
3. Foundational knowledge on working of CPU and Computer Architecture.
4. Knowledge of X86 assembly language.

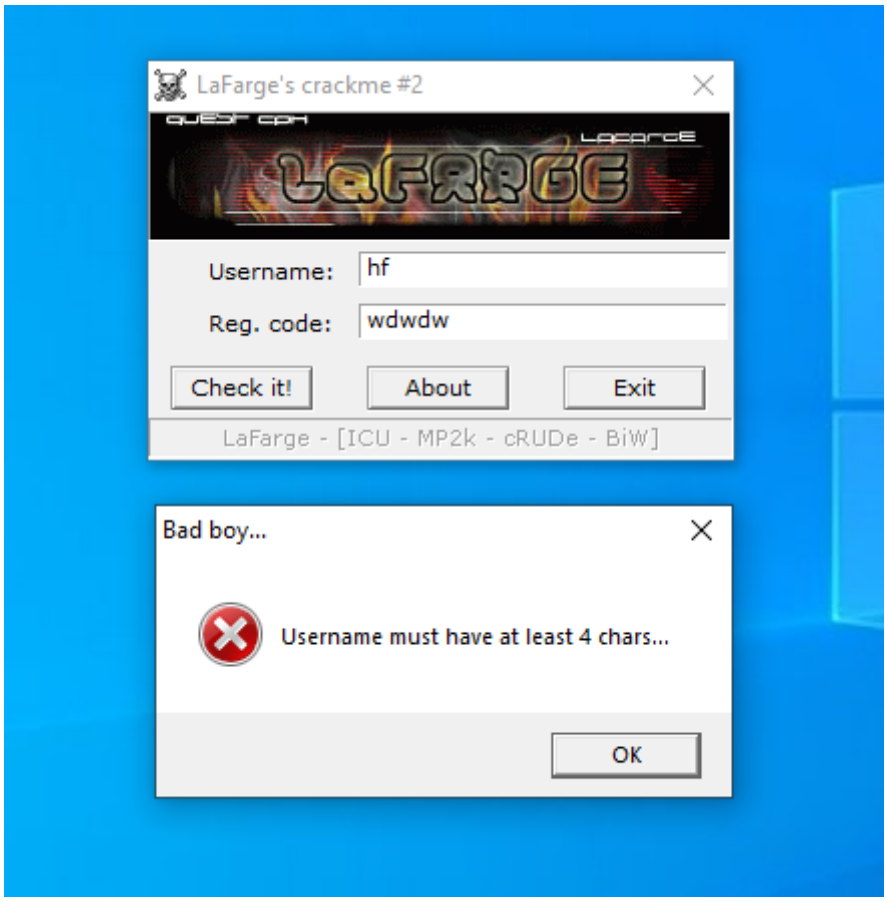
Procedure:-

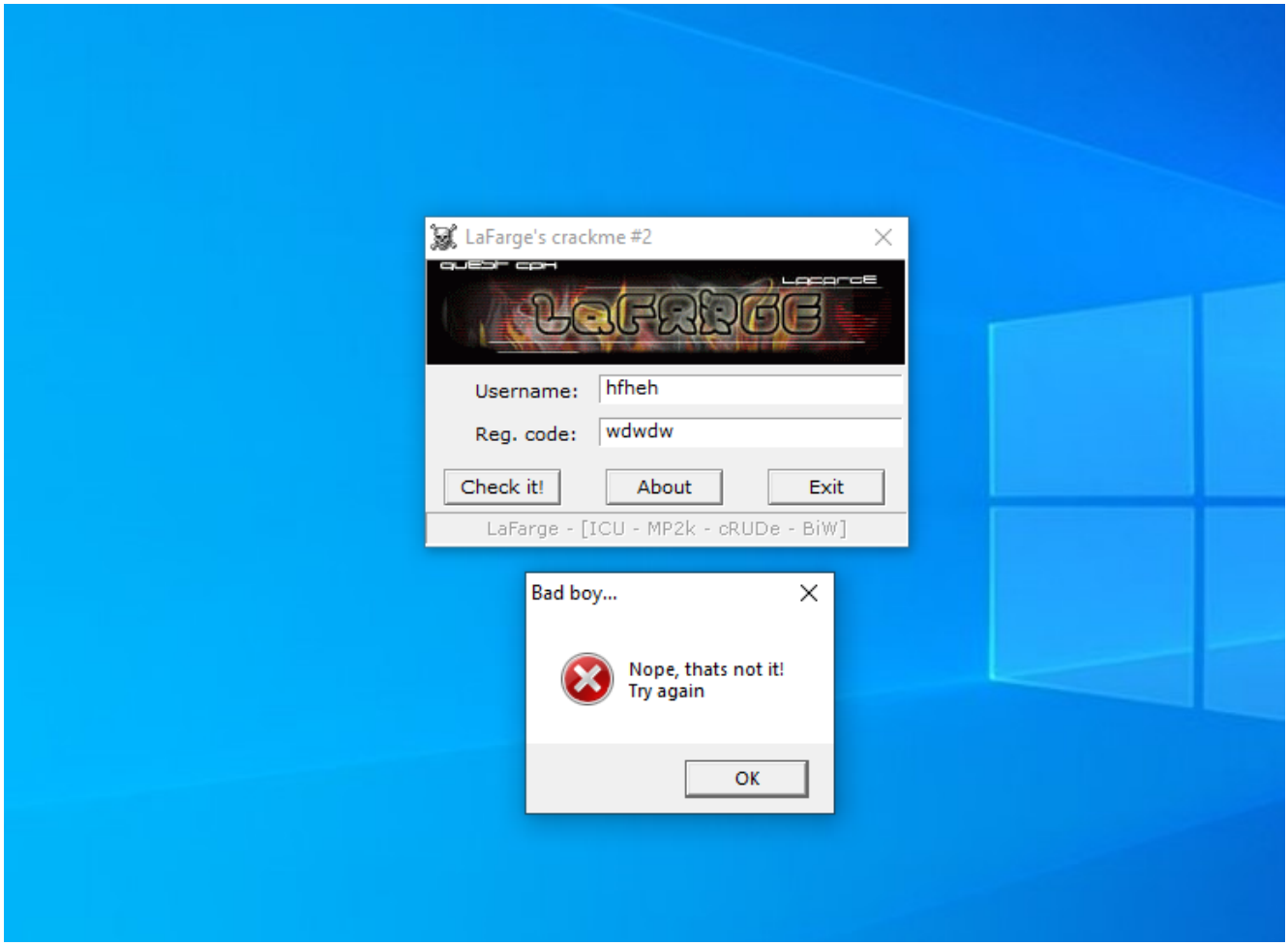
Initially we will open olly debugger.
In this tool we can see various sections.As from the picture we can see the first section is of processor section the second section is of Registers, the 3rd section is of RAM and the fourth section is of

secondary storage.



Now to crack a software First we will check the software for the error messages. Then we will take note of the error messages we need to bypass. In our case we have encountered these two error messages.

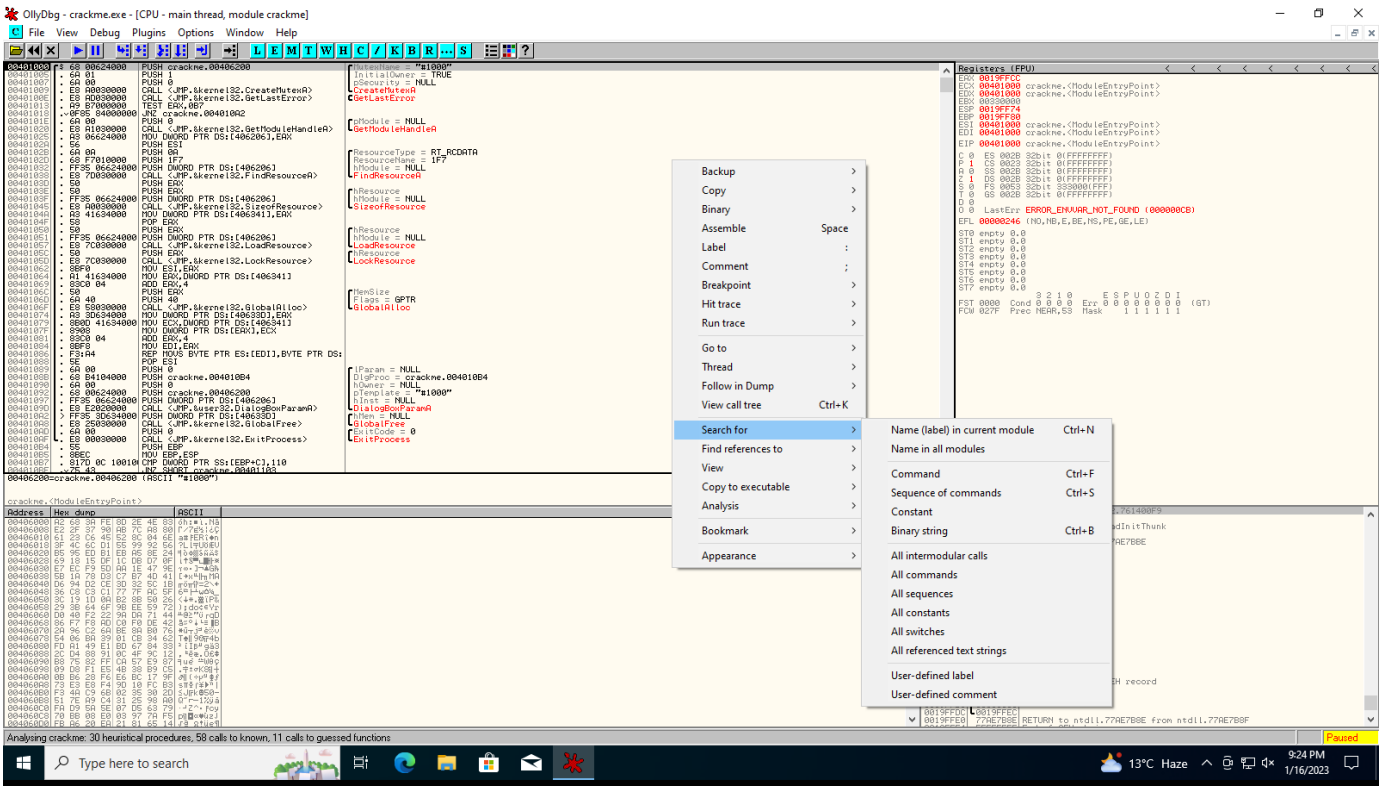
1. 

2. 

Then we will open the software which we are going to crack in the olly debugger. In our case we have crackme software

Then as soon as we open the software in olly debbugger we can see all the disassembly codes that a processor will execute line by line in the processor section of olly debugger.

Now to crack the software first we will have to find the referenced text string which gives the error message.



We can search for all referenced text strings by having a right click and browsing section the search for section.

After finding the referenced text string that we need to bypass we will have to double click onto it which will redirect us to the offset value and the dissasembly code in the processor section.

After finding messages we will have to look for some sort for comparision that will be going on before the error message.

Here We can see we have `CMP` in the assembly code which is used for comparision.

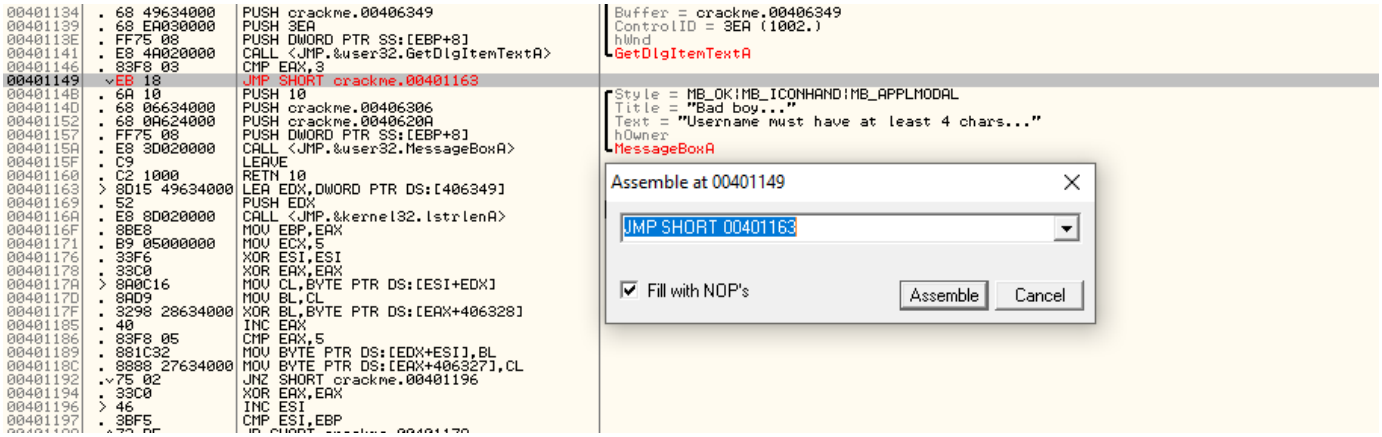
So what the above assembly codes means is that if the Comparision result evaluates to true then only execute the Next line and Jump to the offeset value instructed in the next line otherwise if the comparision result evaluates to false the go ahead and print and error message.

So what we will do is that we will modify the assembly Instruction just below the `CMP` instruction that says JA to JMP. What this will do is that it will always jump to that particular offset value regardless of whatever the comparision result is whether it is true or false.

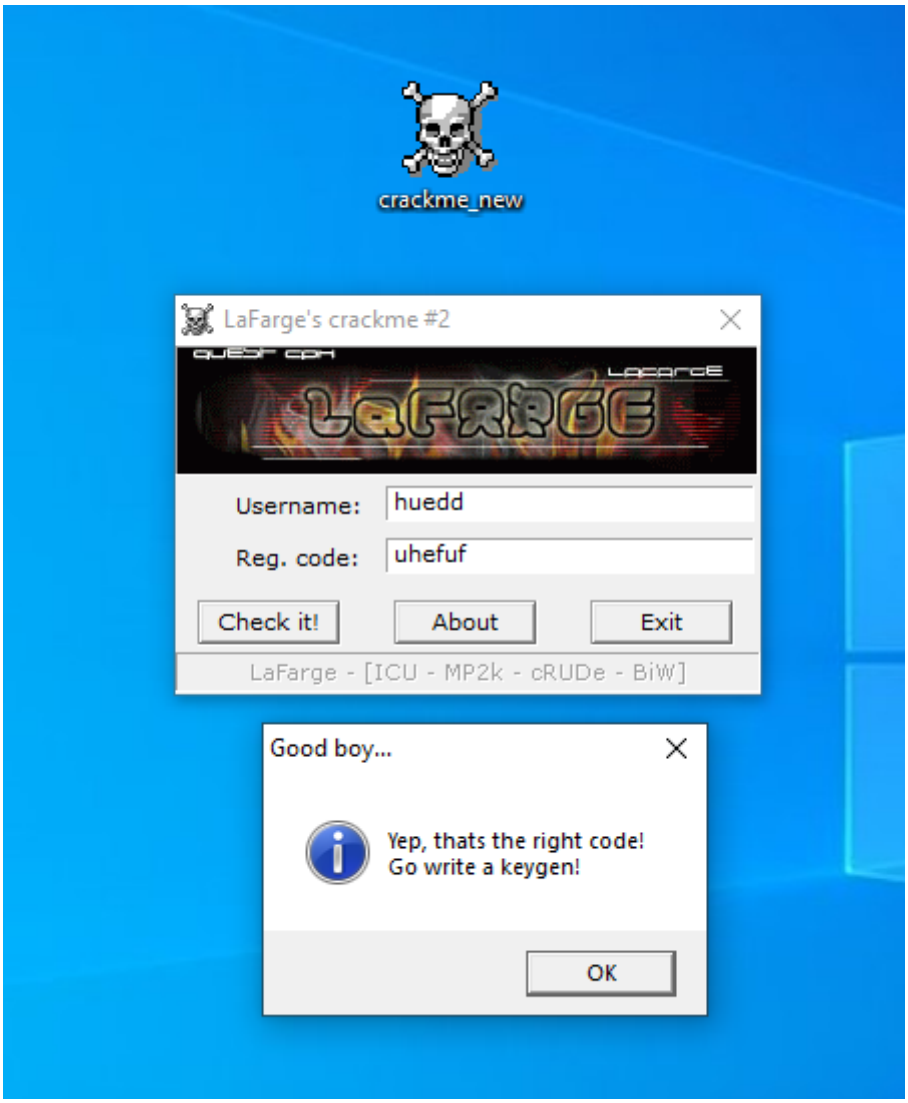
Again we will follow the same steps for the second error message.

```
00401134  . 68 49634000    PUSH crackme.00406349          Buffer = crackme.00406349
00401139  . 68 EA030000    PUSH 3EA                       ControlID = 3EA (1002.)
0040113E  . FF75 08        PUSH DWORD PTR SS:[EBP+8]      hWnd
00401141  . E8 4A020000    CALL <JMP.&user32.GetDlgItemTextA>   GetDlgItemTextA
00401146  . 83F8 03        CMP EAX,3
00401149  .v77 18          JA SHORT crackme.00401163
0040114B  . 6A 10          PUSH 10                        Style = MB_OK|MB_ICONHAND|MB_APPLMODAL
0040114D  . 68 06634000    PUSH crackme.00406306          Title = "Bad boy..."
00401152  . 68 0A624000    PUSH crackme.0040620A          Text = "Username must have at least 4 chars..."
00401157  . FF75 08        PUSH DWORD PTR SS:[EBP+8]      hOwner
0040115A  . E8 3D020000    CALL <JMP.&user32.MessageBoxA>   MessageBoxA
0040115F  . C9             LEAVE
00401160  . C2 1000        RETN 10
00401163  > 8D15 49634000  LEA EDX,DWORD PTR DS:[406349]
00401169  . 52             PUSH EDX                       String => ""
0040116A  . E8 8D020000    CALL <JMP.&kernel32.lstrlenA>   lstrlenA
0040116F  . 8BE8           MOV EBP,EAX
00401171  . B9 05000000    MOV ECX,5
00401176  . 33F6           XOR ESI,ESI
00401178  . 33C0           XOR EAX,EAX
0040117A  > 8A0C16         MOV CL,BYTE PTR DS:[ESI+EDX]
0040117D  . 8AD9           MOV BL,CL
```

And then we will assemble the newly modified instruction into an executable file.



Finally we will check if the newly build executable is working with our modifications or not.



Here we can see it is working perfectly.

That's it we have successfully cracked the crackme software.

Contribution and Impact of Software Cracking on People's Life.

Till now obiviously we can see software cracking can sometimes be very beneficial to us as with the help of it we can use any paid software for free and yeah people always like freebies. How many of you are using windows? Now a days windows comes preinstalled with out modern computers but Do you remember that time, when we use to buy the cracked copy of windows xp, 7 etc. from the market for very cheap and we used to share that copy to our friends. So technically using the same windows copy on multiple system that is only possible with software cracking. Can you now think how much life has been impacted through Software cracking. Every home that had used to have a computer was running

some sort of cracked version of softwares ulimately impacting our daily lives. With the help of software cracking now people don't have to worry about their paid verisons and they can do their daily task tension free and infact now a days cracked softwares also contain some sort of modification which ultimately ease our task than the original one.

So that's it from My side i thank you all for listening me.