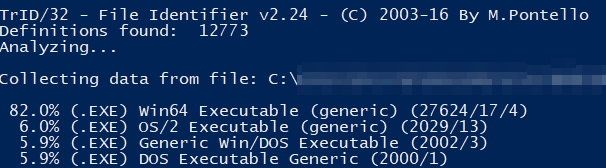
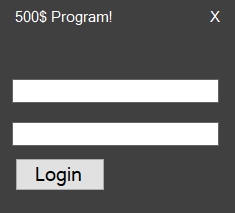
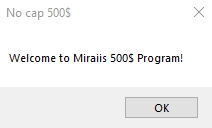
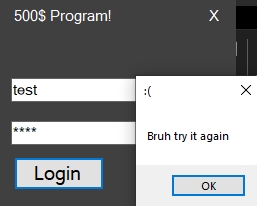
After loading the application into TRiD to ensure it was a windows executable, I threw it to the wolves (dragon) and let Ghidra pull it apart for me. To my surprise, many bad instructions, lacking code flow and overall unusual viewing led me to consider that it isn't standard C/C++ but coded within .net C#



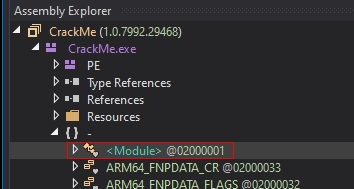
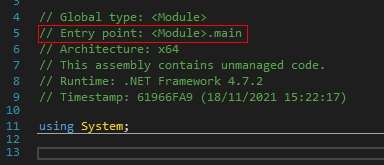
Running the application gave me the impression that it was created by normalised forms with a simplistic appearance, opposed to some fancy C/C++ coded GUI.



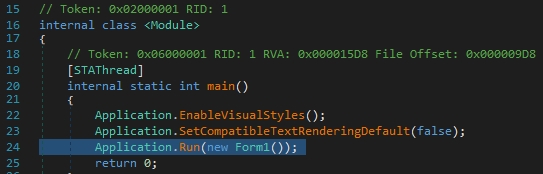
Inputting typical fake data of test - test for the proposed username and password, I was politely informed that I must try again. Fine!



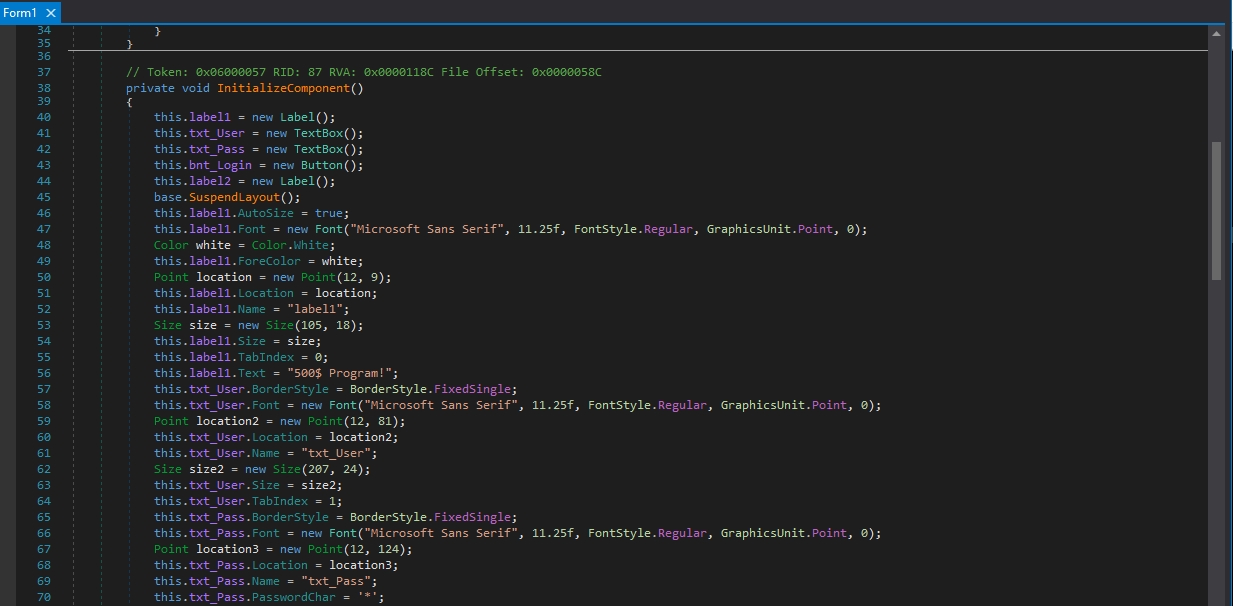
Loading up DNspy to confirm my suspicions, the application structure was laid bare for me to look over. Checking over the CrackMe.exe data, the entry point was infact main, not entry that Ghidra had loaded up. Clicking on <Module>.main takes you to the real entry point of the application that the user sees, opposed to the setup of the application within windows.



This is where I had to resist the urge to charge off ahead thinking I knew what I was doing, as I'd already hit a blocker with Ghidra. Within the first 24 lines, I'd already seen something that stood out, so checked out what this "Application.Run(new Form1())" meant, given .net works by creating windows called forms as a GUI by clicking on Form1.

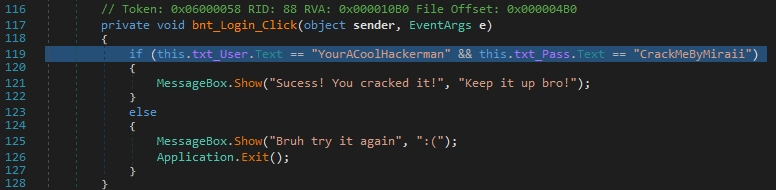


Now dependant on the font size on the monitor, or screen size, or various other factors that I can't fathom, I could see that on line 47 there's a line that initialises font in preparation for displaying text on the screen. Upon scrolling down, I then see the 500$ Program! line that is apparent whilst on the login screen on the application. This led me to believe that I'm on the right lines if nothing else.



At the bottom of the above picture, you can see the lines that setup the password field to replace cleartext into \*\*\* replaced text, imitating a password. Nice!

Now this is where the application has been created with the ending first, then the introduction as an afterthought due to the way that the ID's have been numbered within DNspy. Given it's a crackme, this isn't overly surprising, as the creator possibly wanted to test that the code worked first, before giving it an aesthetic look.



From the above, we can see that the answer to the login is hard coded in clear text thankfully, so no trickery required to figure out. However, this could be a red herring, so loading up the challenge again and inputting the username and password to confirm whether we get the "Success!" message box or not...... And we do!

