DAY18_advent_of_the_cyber

Day 18: The Bits of Christmas - Story:

"Silly Santa...Forgetting his password yet again!" complains Elf McEager. However, it is in fact Elf McEager who is silly for not creating a way to reset Santa's password for the TBFC dashboard.

Santa needs to get back into the dashboard for Christmas! Can you help Elf McEager reverse engineer TBFC's application to retrieve the password for Santa?!

Lost in the depths that is .NET? Follow along with John Hammond for today's task.

18.1. Getting Started:

Before we begin, we're going to need to deploy two Instances:

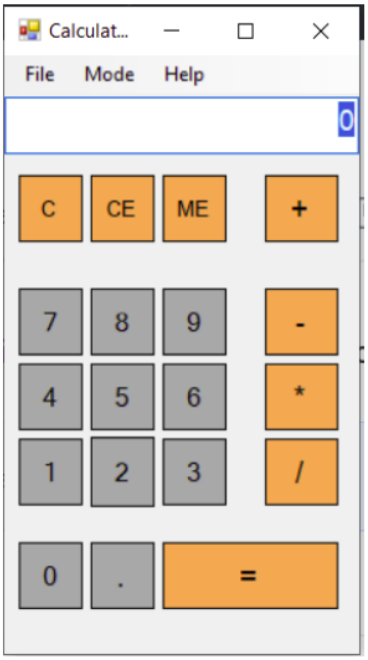
1. The THM AttackBox by pressing the "**Start AttackBox**" button at the top-right of the page.
2. The vulnerable Instance attached to this task by pressing the "**Deploy**" button at the top-right of this task/day

Made with ❤ by CMNatic

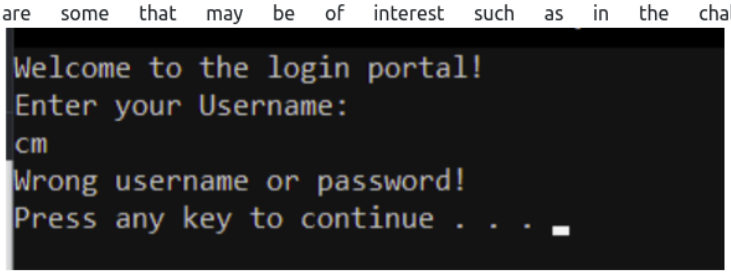
You got your hands dirty with everything that is radare2 yesterday. Today, however, we're going to be taking a look at a more interactive approach of disassembling an application.

Due to its compatibility and long history, the .NET Framework is a popular platform for software developers to develop software with. Anything Windows or web, .NET will cover it.

For example, I developed my answer to Microsoft's Calculator in .NET:

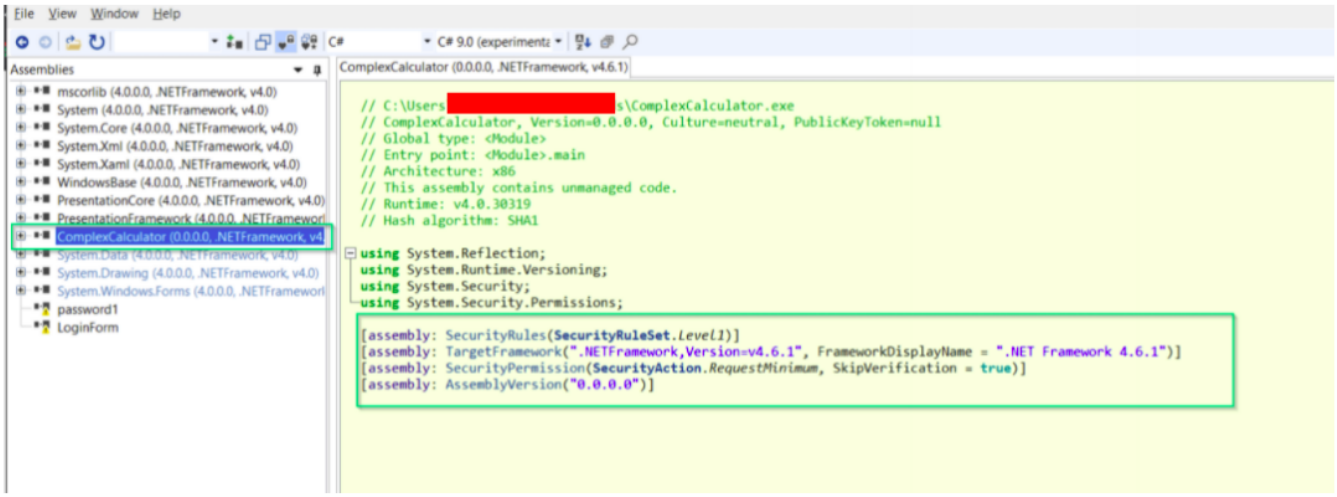


This is quite a trivial use of .NET, but hey, it works (trust me on this one okay?). Whilst you may not want to take a look behind the code of this application, there are some that may be of interest such as in the challenge today. Let's take a look at the application below:

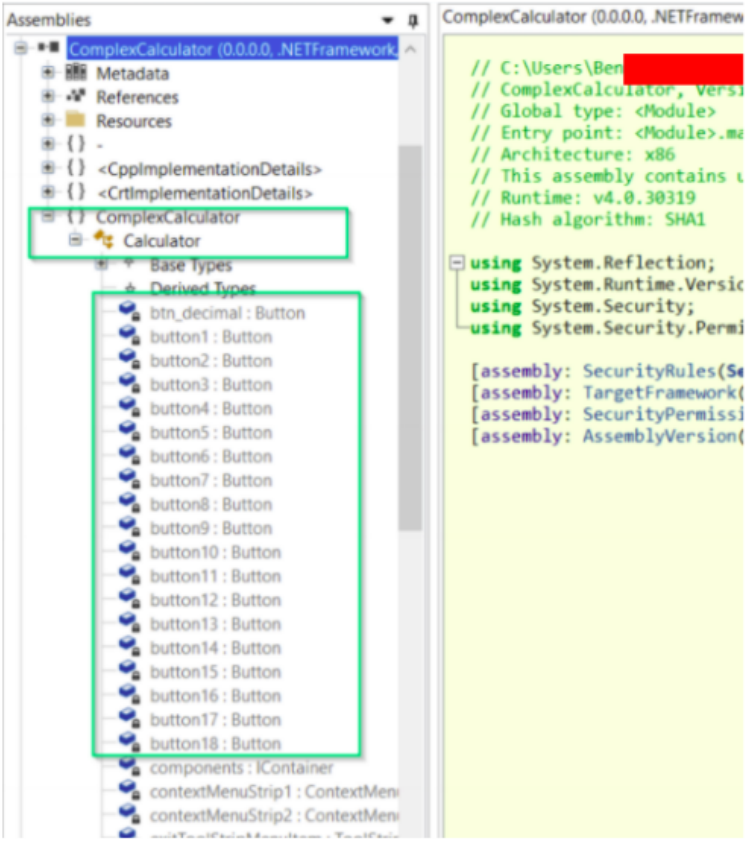


When running the application, we are asked for an input (in this case a Username). This begs the question, how does the application know what username/password is right or wrong? The application must know the answer...Applications that are created using the .NET framework can be disassembled using tools such as ILSpy or Dotpeek.

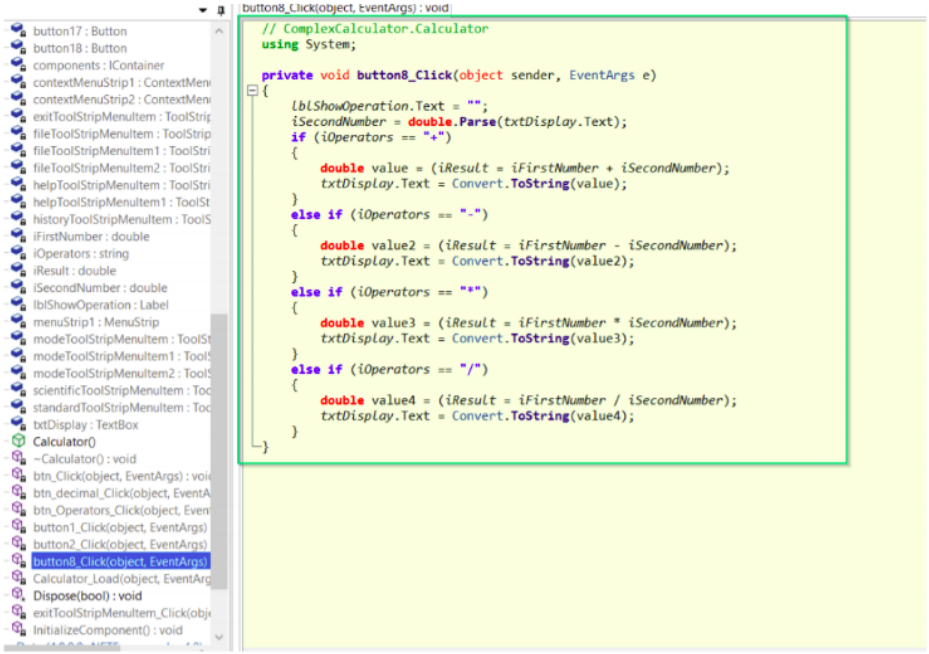
Loading our calculator application into ILSpy verifies that it is indeed a .NET application:



After expanding some of the resources, we can see references to elements of the application such as buttons, labels and the likes:



When looking through the objects, ILspy has helpfully been able to recreate what some of the source code behind the application is:



Because it's a calculator, we can see the c++ code that checks for mathematical operators (plus, minus, multiply and divide). Looking through other objects reveals similar code (of that we'd expect of a Calculator at least).

18.3. Challenge:

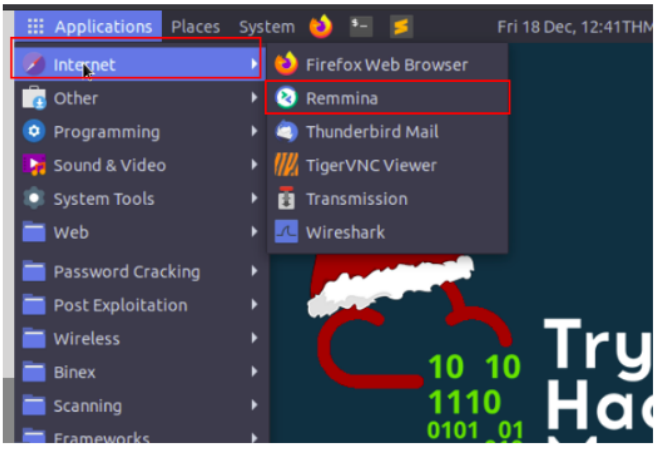
Deploy the instance attached to this task and log in using the Remote Desktop Protocol (RDP). Open the application "TBFC_APP.exe" on the Desktop and enter the correct password!

You can use "Remmina" on the TryHackMe AttackBox to connect to the instance with the following credentials, or any RDP client such as Microsofts if you wish to connect to the TryHackMe VPN:
IP Address: `10.10.95.71`

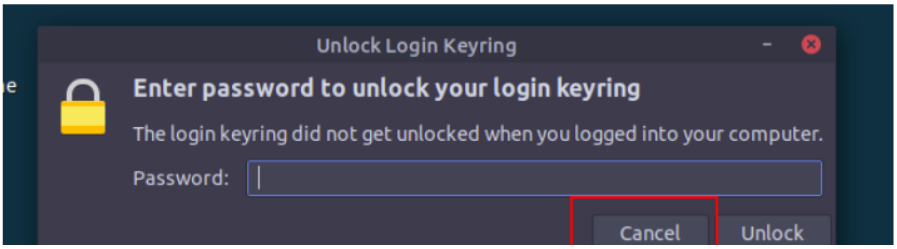
Username: cmnatic

Password: Adventofcyber!

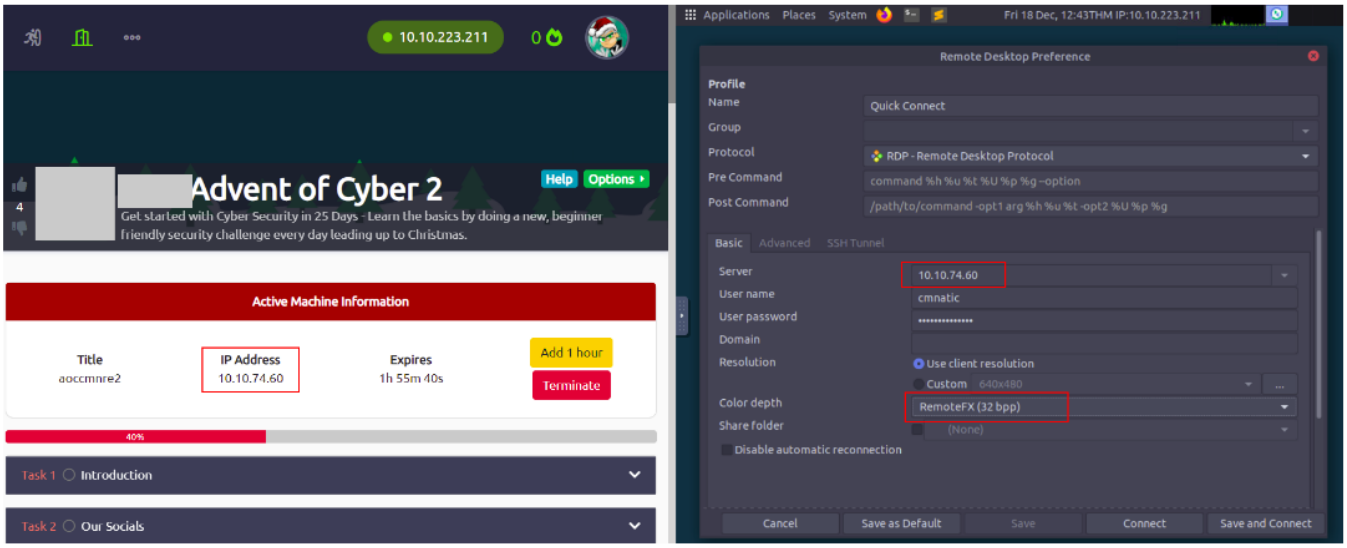
1.Navigate to the "Applications" tab on the AttackBox where "Remmina" is located in the "Internet" sub-menu.



2. Reminna will ask you for a password to save sessions, we can safely press "Cancel":



3. Now fill out the IP address of the target Instance that you have deployed, input the Username and password provided and set your "Color depth" to "RemoteFX (32 bpp) like so:



IP Address: `10.10.95.71`

Username: cmnatic

Password: Adventofcyber!

*As this is a Windows box, **please allow a comfortable five minutes for it to fully set up**. Grab some water (into a container, please, unless you're a water bender. If you are in fact, why are you reading this task?) and do a quick posture check.*

Want to get more hands-on with disassembling applications on Windows? Check out my Malware Analysis primer:
1. Malware Analysis Primer - learning and visualising the characteristics of malware.
2. MAL: REMnux - The Redux - Using REMnux to analyse malicious PDFs and the memory dump of a machine infected with the Jigsaw Ransomware.

Open the "TBFC_APP" application in ILspy and begin decompiling the code

No answer needed | Correct Answer | Hint

What is Santa's password?

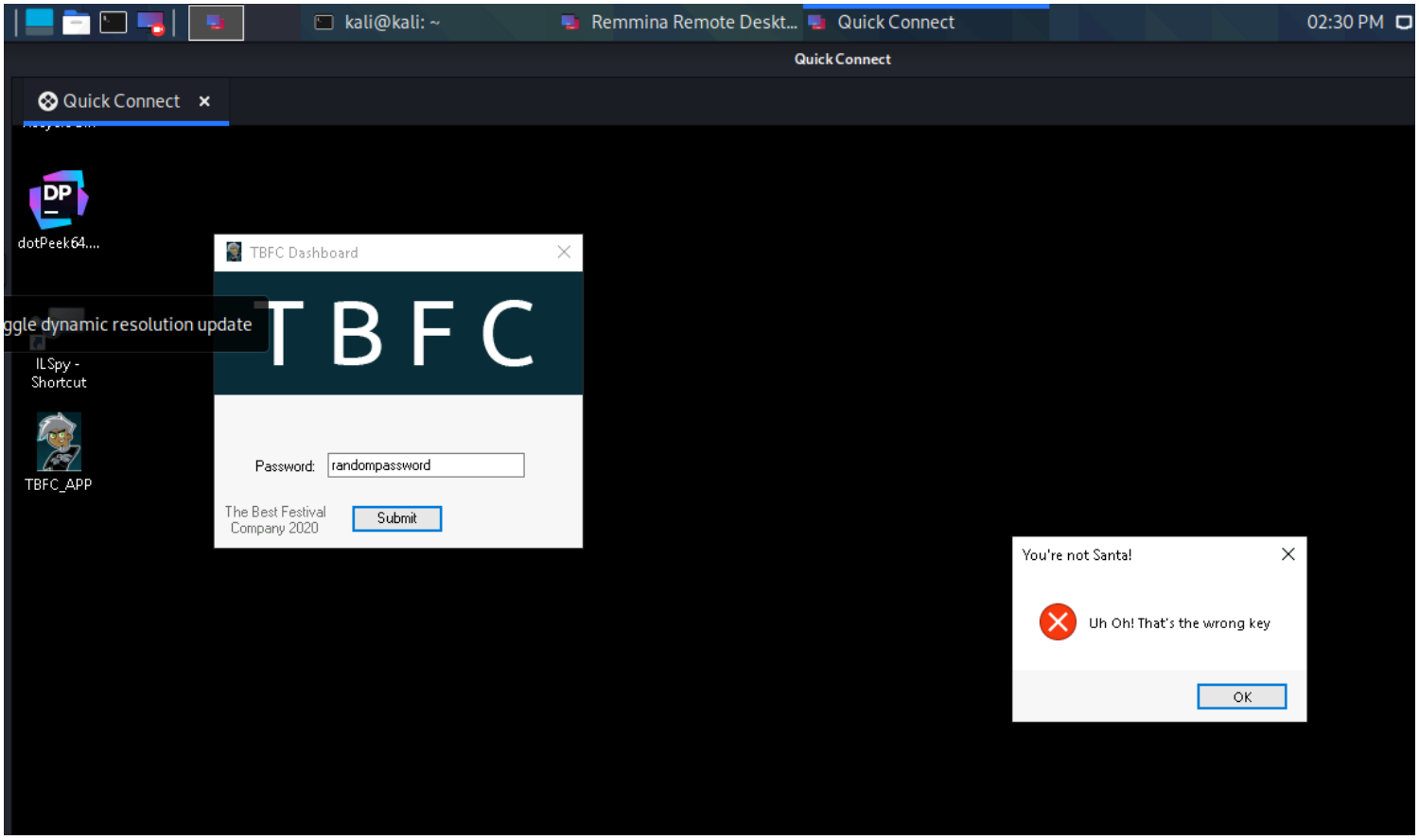
santapassword32 | Correct Answer | Hint

Now that you've retrieved this password, try to login...What is the flag?

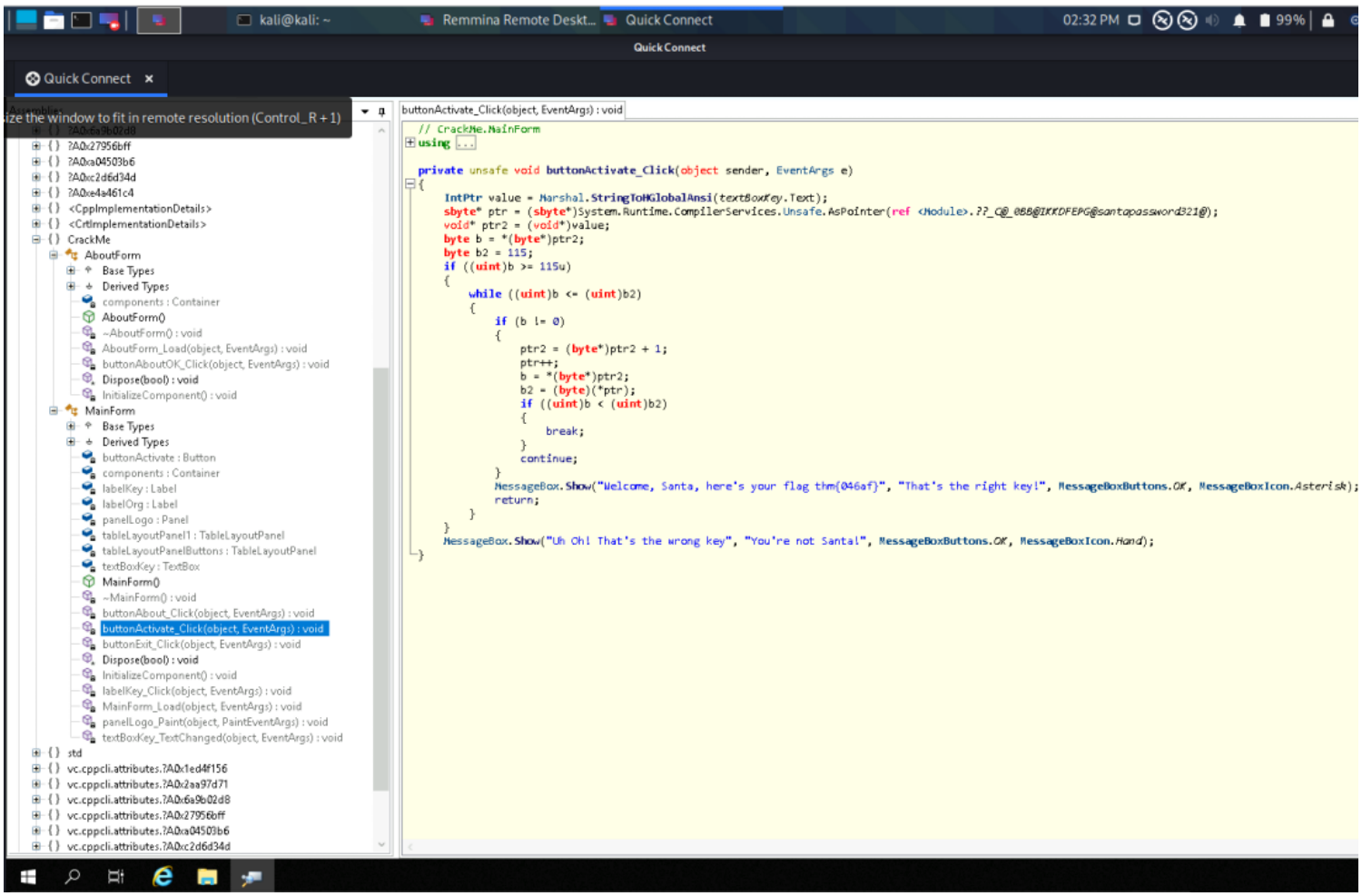
thm{046af} | Correct Answer

solving the challenge:

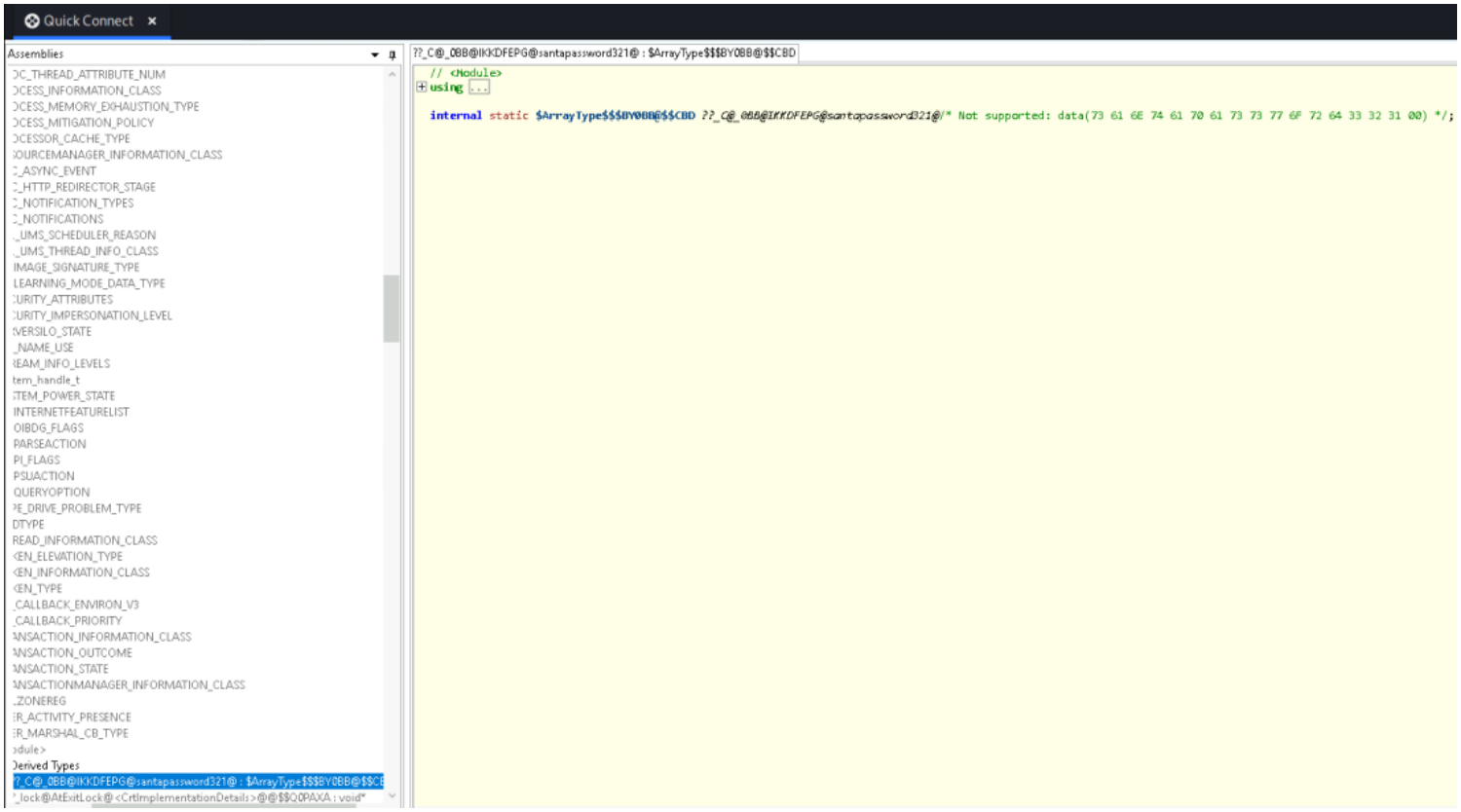
after login through RDP in remote desktop, we try to use a random password



Loading TBFC_APP in ILspy application:



we can see both the password and flag while analyzing the application's code , but let's analyze password a lil bit more coz it could have been a variable too:



Passing the hex in cyberchef web app:

Last build: A month ago

Recipe

From Hex

Delimiter
Auto

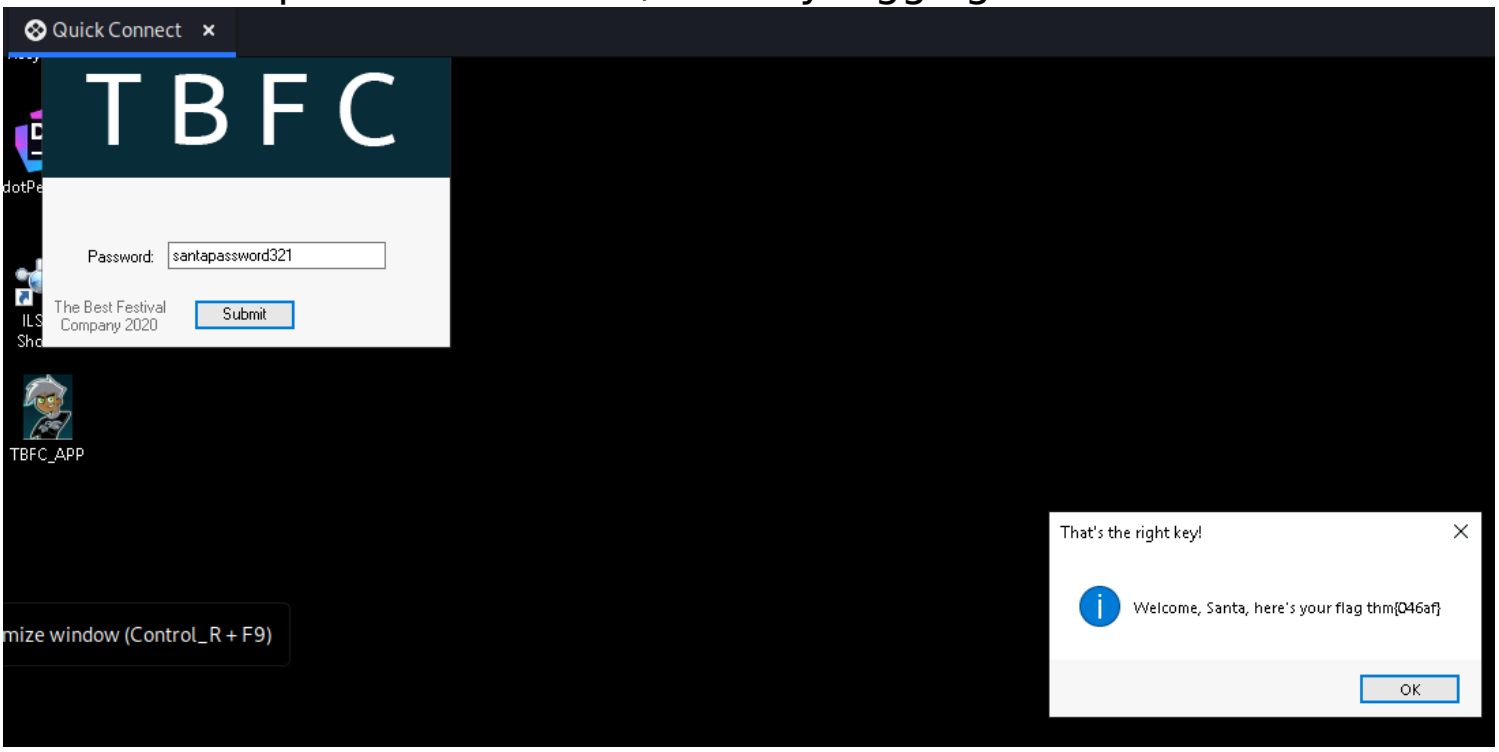
Input

```
73 61 6E 74 61 70 61 73 73 77 6F 72 64 33 32 31
```

Output

```
santapassword321
```

we have the password with us, let's try logging in :



we are successfull :)