

राईकवाCTF

CHALLENGE NAME: [Safe Box]

DEV: [Soham]

CATEGORY: [Reverse]

LEVEL: [Hard]



2025

Challenge Description:

There are many ways, but the choice is yours.

File: [Safe Box.zip](https://drive.google.com/file/d/1YSqk_v77QZo61ePBcukLnFxLsXYbQK8C/view?usp=drive_link) (https://drive.google.com/file/d/1YSqk_v77QZo61ePBcukLnFxLsXYbQK8C/view?usp=drive_link)

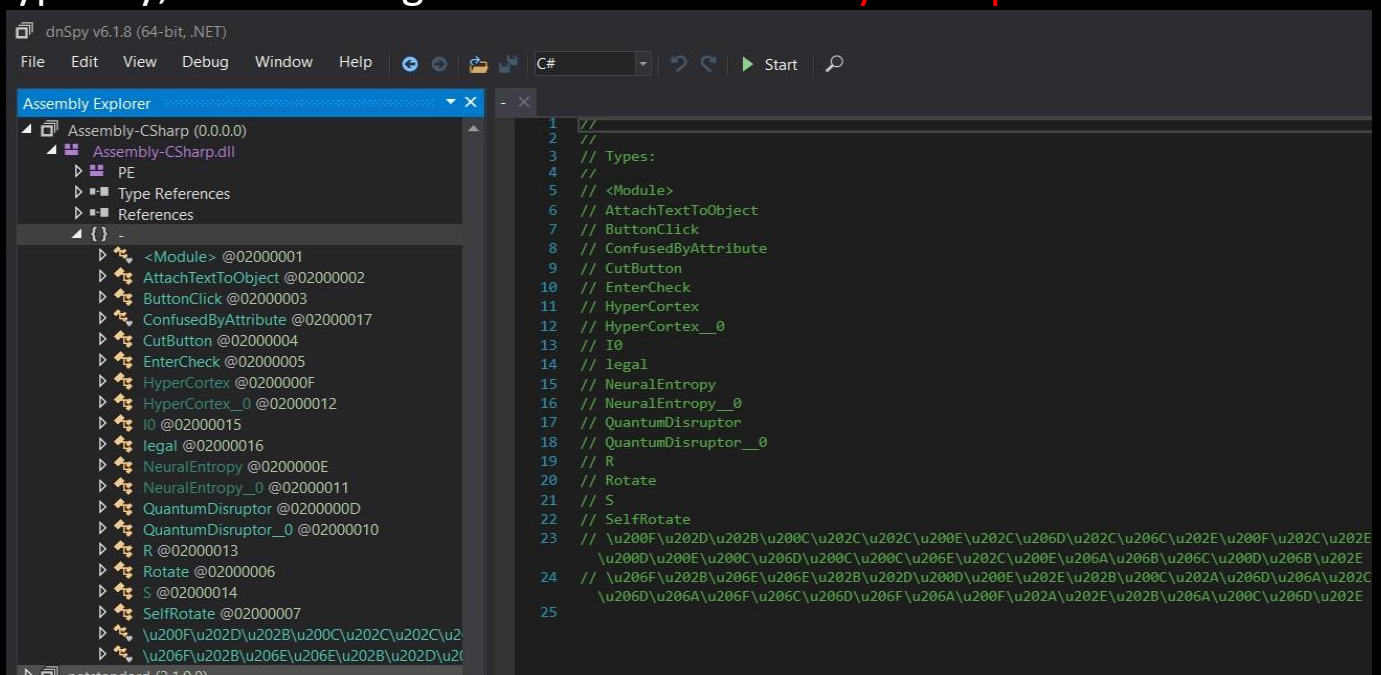
Solution:

Upon opening `Safe.exe`, you must enter the correct PIN or password to unlock the Safe Box and retrieve the flag. Additionally, you can find PIN, bypass PIN verification, remove the door, or directly access the flag object or the hardest way- you can reverse PIN generation to get it.

Let's start,

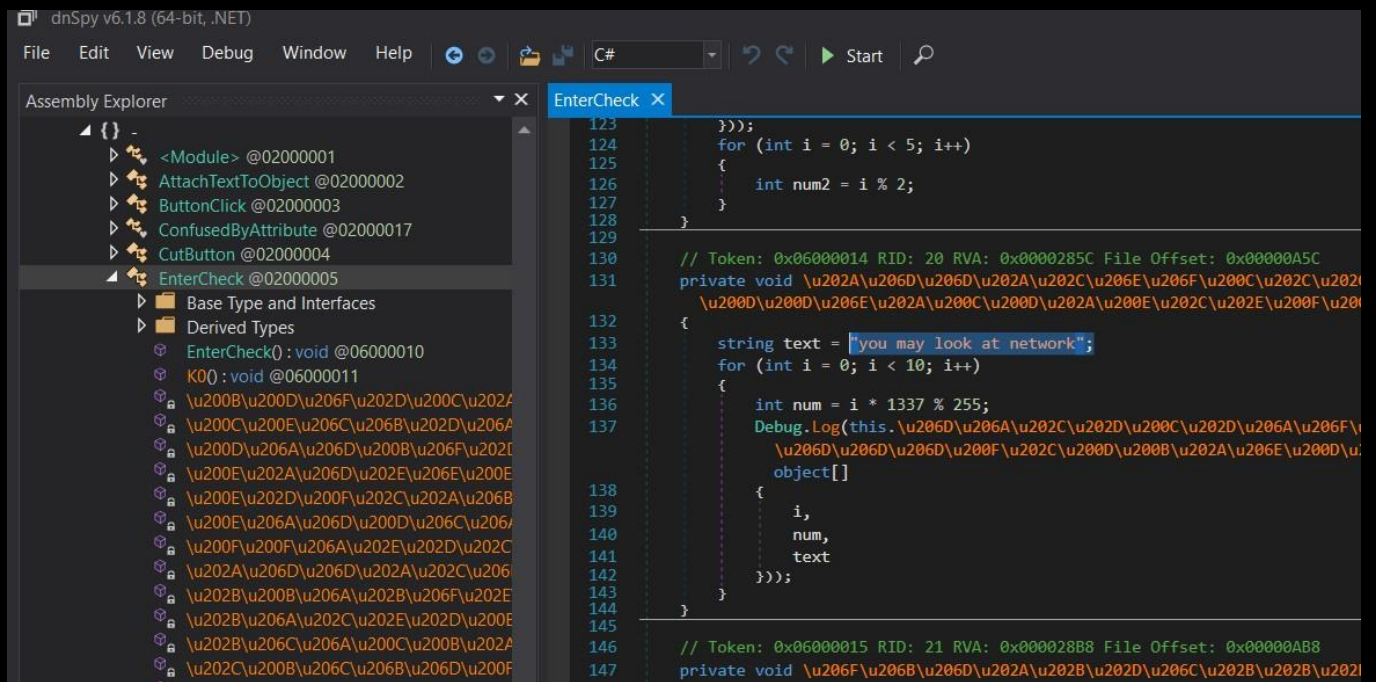
1. Since Unity stores game logic in DLL files, navigate to the **Managed folder**, where multiple DLL files are located.

To determine which DLL contains the game logic, you can ask an AI. Typically, the main logic is stored in **AssemblyCSharp.dll**.



Upon opening this file in [dnSpy](#), we see that it is obfuscated but class names are still visible.

However, the **strings** in this .dll were left **unobfuscated**, so take a moment to read them.



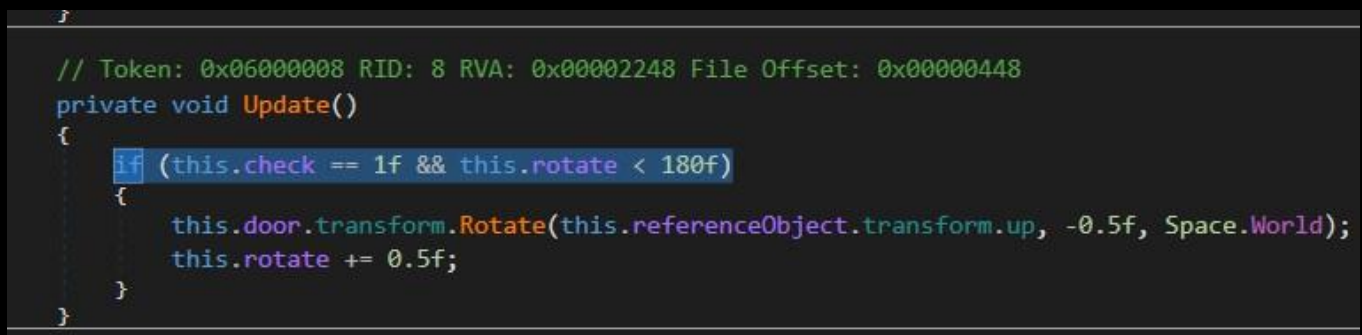
The screenshot shows the dnSpy v6.1.8 (64-bit, .NET) interface. On the left, the Assembly Explorer displays the 'EnterCheck' class. The main editor shows the deobfuscated C# code for the 'EnterCheck' class. The code includes a 'private void EnterCheck()' method that contains a loop and a 'Debug.Log' statement. The string 'you may look at network' is visible in the code.

```
// Token: 0x06000014 RID: 20 RVA: 0x0000285C File Offset: 0x0000A5C
private void EnterCheck()
{
    for (int i = 0; i < 5; i++)
    {
        int num2 = i % 2;
    }

    // Token: 0x06000015 RID: 21 RVA: 0x000028B8 File Offset: 0x0000AB8
    private void Update()
    {
        string text = "you may look at network";
        for (int i = 0; i < 10; i++)
        {
            int num = i * 1337 % 255;
            Debug.Log(this.ToString() + " " + num + " " + text);
        }
    }
}
```

something suspicious: "You may look at network." This suggests checking for any network-related files.

In the Managed folder, there is a file named **UnityEngine.NetworkUtils.dll**. Opening this file in dnSpy, we find that it contains deobfuscated game logic. At this point, you can either feed it to an AI for analysis or examine it manually. Within the code, an if statement checks the password. In EnterCheck Class inside Update() method.

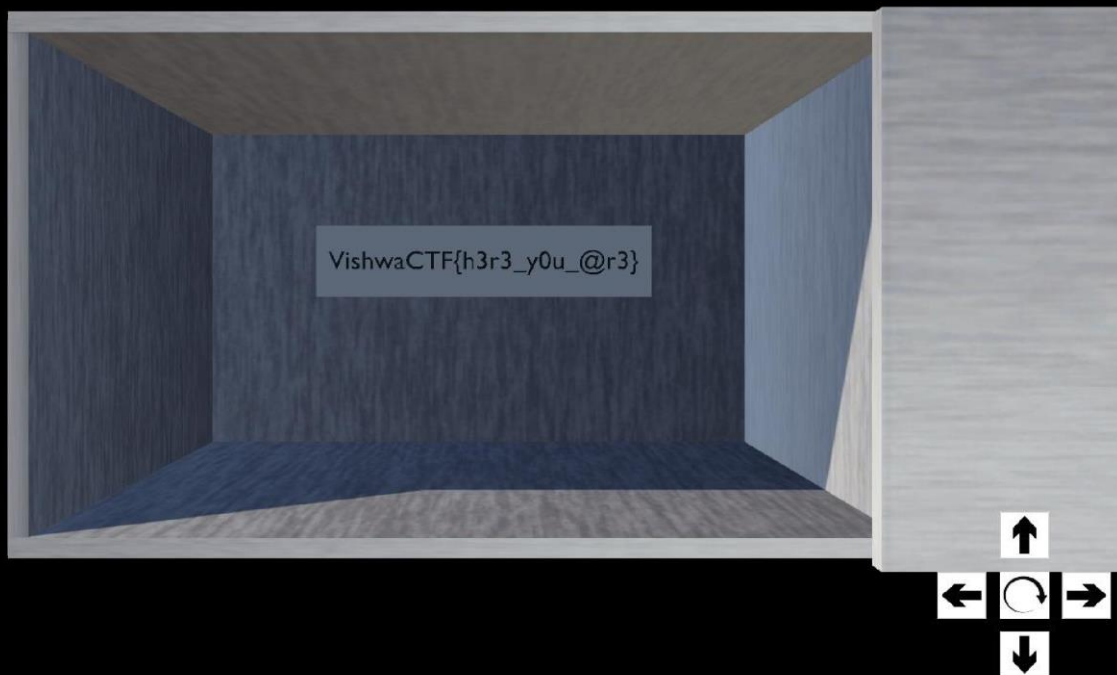


The screenshot shows the 'Update()' method of the 'EnterCheck' class. The code includes an 'if' statement that checks 'this.check == 1f' and 'this.rotate < 180f'. If the condition is true, it calls 'this.door.transform.Rotate' and increments 'this.rotate' by 0.5f.

```
// Token: 0x06000008 RID: 8 RVA: 0x00002248 File Offset: 0x0000448
private void Update()
{
    if (this.check == 1f && this.rotate < 180f)
    {
        this.door.transform.Rotate(this.referenceObject.transform.up, -0.5f, Space.World);
        this.rotate += 0.5f;
    }
}
```

Modifying this statement to **if(this.rotate < 180f)** and saving the DLL allows you to bypass the password check.

Now, reopen the Safe.exe and get the flag.



2. Now this process is quite long but interesting.

The Safe Box allows only three password entry attempts. If all attempts are incorrect, it locks permanently. To reset the counter and gain three more attempts, delete the reset.exe file in the Safe_Data folder.

But resetting the counter will not help a lot.

Process shown in [Video](#):

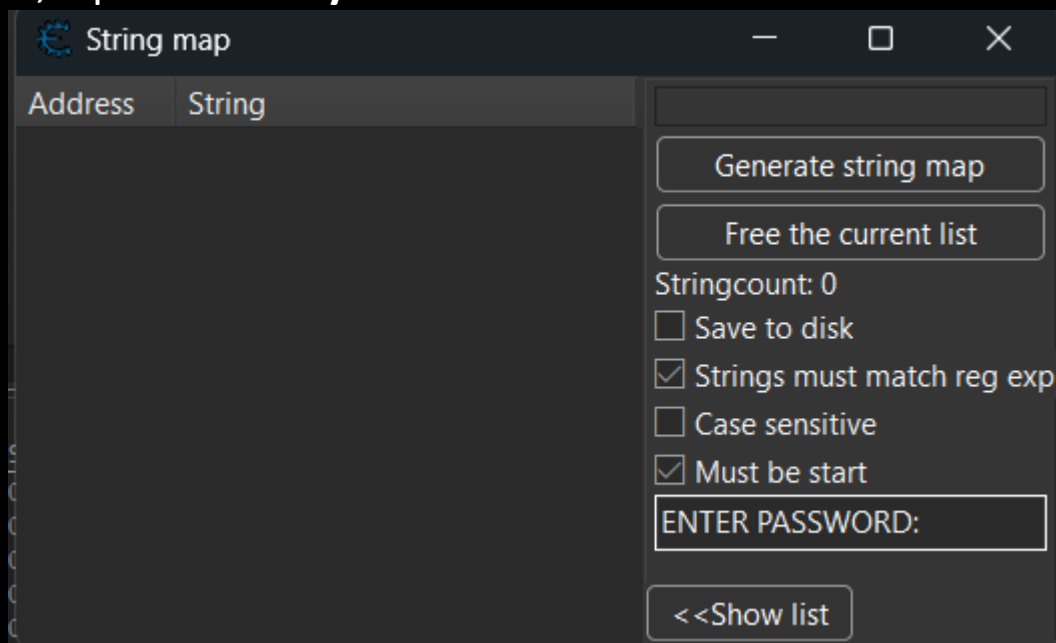
<https://drive.google.com/file/d/1dHXvKrx2DFvpwmy5a1FTAz6E1MRuxu01/view?usp=sharing>

1. So, Open [Cheat Engine](#) and attach it to the game process.
2. Enter your current number of tries in the **Value** field. (initially 0)
3. Click **First Scan** → Enter a wrong password → Update the tries count and click **Next Scan**.
4. Repeat until you find the correct address. (in this case 3 times)
5. Right-click the **tries counter address** → **Change Value** to 0. ↓
6. Freeze the value to prevent it from increasing. ↓

7. Now, you can enter unlimited passwords! – (if u want to brute force.)

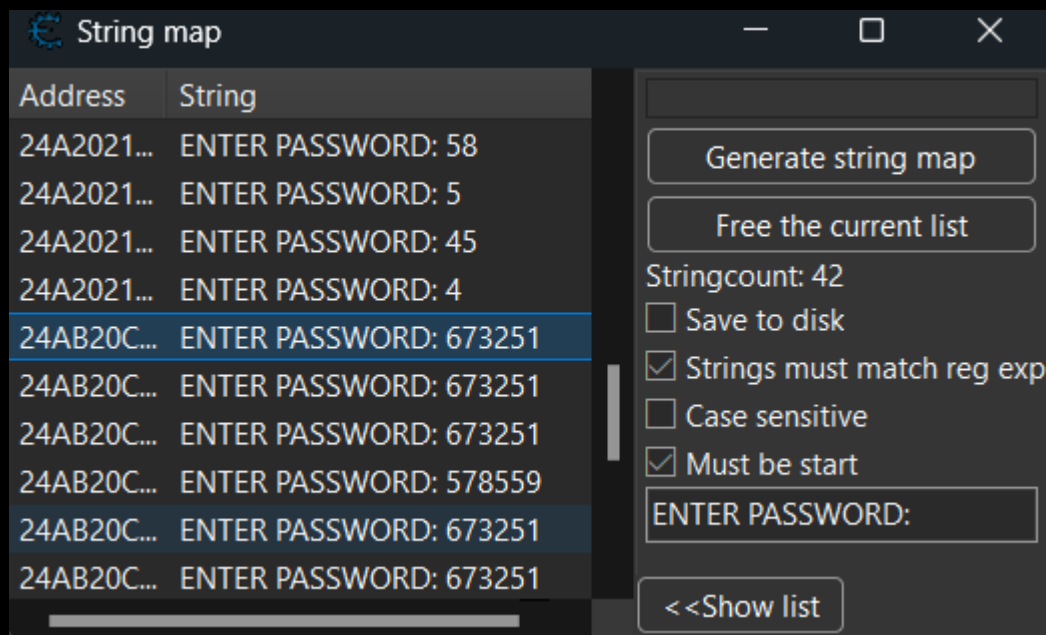
8. No need to “Find out what writes to this address”.

9. Now, Open **Memory View** → Click **VIEW** → **ALL STRINGS**.



10. And enter “ENTER PASSWORD:” -> Generate string map -> Show list.

11. You will see few PINs, try them and you will get correct one which is “673251”



12. Get the Flag by entering correct password.

Flag:

VishwaCTF{h3r3_y0u_@r3}

Another way you can use Asset Ripper and extract the project in unity
And locate the flag or door object and drag it to see the flag.