Category

Reversing

Challenge description

Flappy Bird is soooo easy, so we decided to make it a little bit harder (and longer) :P. In fact, we may have gone a little too far. Have fun!

Hints

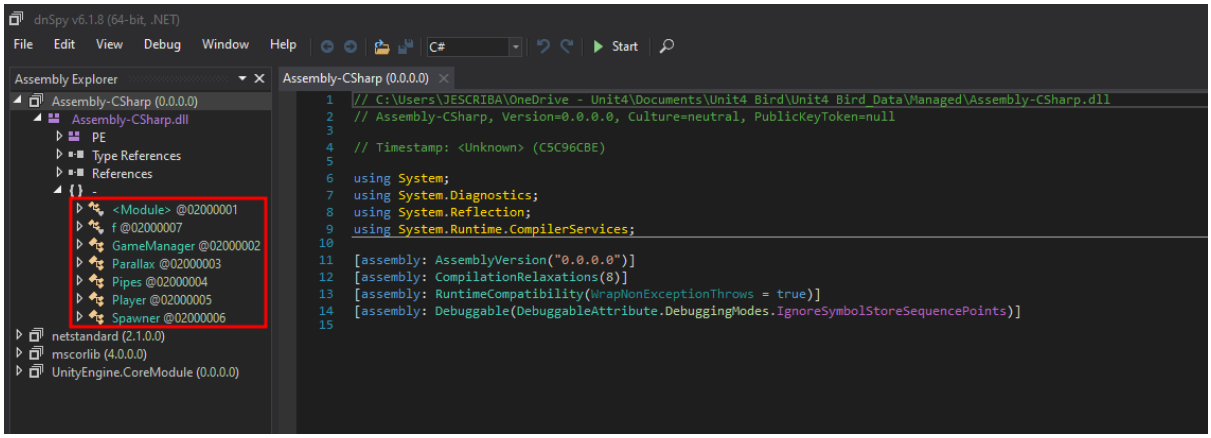
- {LEVEL 1} Have you looked inside the managed data? :P
- {LEVEL 2} Maybe you want to use a decompiler for Assembly.CSharp.dll, there is plenty of interesting things there.
- {LEVEL 3} You can find GameManager inside Assembly.CSharp.dll, I think I left there a method to win the game when you get to a specific amount of points…

Write Up

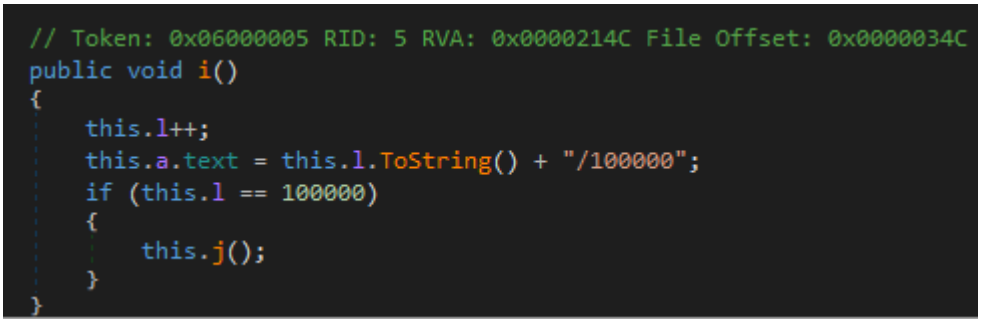
In this challenge, we have a Flappy Bird made in Unity. The objective is to achieve the number of points shown on screen, but it's hard to do it because it's a very high number.

To avoid playing the game to achieve the flag, we need to reverse the code of the game using any software that allow us to decompile assemblies and analyze the code. For example, we can use **dnSpy.**

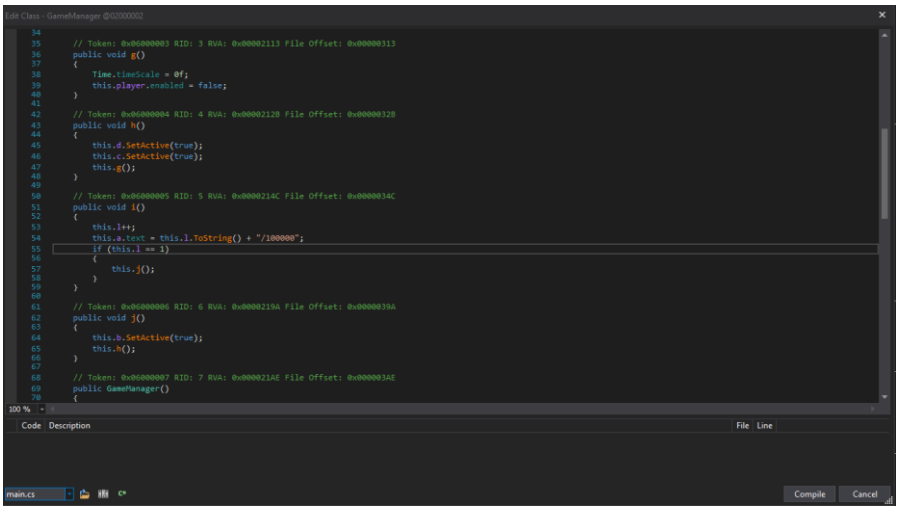
In the Data folder of the game, we will see all the assets and DLLs. You will find the logic of the entire game inside **Assembly-CSharp.dll**. To take a look inside that DLL, we will open it with dnSpy.



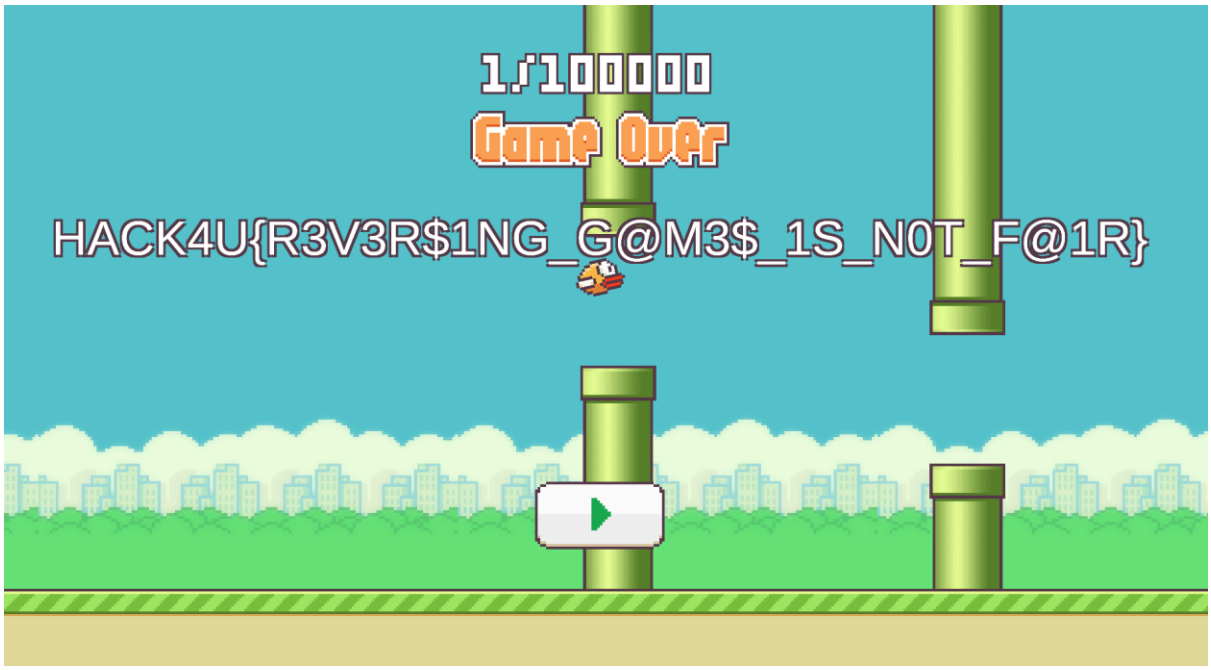
The code related to winning the game and showing the flag is in GameManager. The code is obfuscated, but if we take a look inside it we will find the following method:



When we enter that if statement, we win the game and we will show the flag. So, to show the flag, we will change the value from 100000 to 1. By that, we will automatically see the flag when we get into the first pipe inside the game. This is only one way to do it but there's plenty of them, like setting the text for the flag to True, changing the number of points when we pass a pipe, etc…



When we're finished modifying the game, we will compile it (dnSpy will do it for you) and you will see your changes applied in the game.



Flag

HACK4U{R3V3R$1NG_G@M3$_1S_N0T_F@1R}