Tutorial: How to deobfuscate Assembly-CSharp.dll

Table Of Contents

- 1 Deobfuscation
- 2 Fixing "ResolutionScope is null"
- 3 Notes

Requirements:

- de4dot (specific version from Senko's dev repo download here)
- dnSpy (download)

1 Deobfuscation

- 1. Copy-paste EscapeFromTarkov_Data/Managed/Assembly-CSharp.dll to where you extracted de4dot (same folder where de4dot-x64.exe is).
- 2. Drag and drop the Assembly-CSharp.dll on top of de4dot-x64.exe.
- 3. You should see the following:



Next, you'll need to locate a token to finish cleaning the assembly.

4. Open the cleaned Assembly-CSharp.dll file in dnSpy (File > Open... OR Ctrl+O).

5. In the search tab¹, choose to search for Number/String² and set the search scope to Selected Files³. Then, type =⁴ and you will get a bunch of results. We're looking for a method called method 0⁵, which should be inside of a nested class. Double click it.

Search	· · · · · · · · · · · · · · · · · · ·	×
= 4	📀 Options Search For: 🗉 Number/String 🔁 Selected Files 🕇	
HeightBias	🕆 TOD_FogParameters	*
ITEMS_COUNT_TEXT_FORMAT	🔩 GClass1340	
ITEMS_REQUIRED_NUMBER_TEXT_FORM	AT 😪 GClass1340	
🗗 LessOrEqual	FT.Quests.ECompareMethod	
m_TargetBleachColor	🔩 PrismEffects	
© method_0 5	🔩 Class2054.Class2055	
MinimumHeight	🙀 TOD_LightParameters	
🗗 MoreOrEqual	₽ EFT.Quests.ECompareMethod	
🥥 mute	🔩 BehaviourMachine.SetAudioMute	▼
Search Analyzer		

6. From there, you should be able to locate a method called smethod_0, near the top of the class. You want to copy the Token value, displayed above the method's definition:

// Token: 0x0600D49F RID: 54431 RVA: 0x00127B77 File Offset: 0x00125D77 public static string smethod 0(int int 0) return (string)((Hashtable)AppDomain.CurrentDomain.GetData(Class2046.string 0))[int 0];

7. Now, create a .cmd file in the de4dot directory with the following contents:

de4dot-x64.exe --un-name "!^<>[a-z0-9]\$&!^<>[a-z0-9]_..*\$&![A-Z][A-Z]\\$<>.*\$&^[a-zA-Z_<{\$][a-zA-Z_0-9<>{}\$.`-]*\$" "Assembly-CSharpcleaned.dll" --strtyp delegate --strtok "YOUR TOKEN HERE" pause

- 8. Replace the YOUR TOKEN HERE part with the token you copied (should look something like this: -- strtok "0x0600D49F").
- 9. Run your newly created .cmd file if you copied the correct token, you should see this screen:

C:\WINDOWS\system32\cmd.exe			×
\de4dot>de4dot-x64.exeun-name "!^<>[a-z0-9]\$&!^<>[a-z0-9]*\$&![A-Z ^[a-zA-Z_<{\$][a-zA-Z_0-9<>{}\$.`-]*\$" "Assembly-CSharp-cleaned.dll"strtyp delegatestrtok "0x0600D49F"][A-Z]\\$<>.	*\$&
de4dot v3.1.41592.3405			
Detected Unknown Obfuscator (\de4dot\Assembly-CSharp-cleaned.dll) Cleaning \de4dot\Assembly-CSharp-cleaned.dll Renaming all obfuscated symbols Saving \de4dot\Assembly-CSharp-cleaned.dll			
ERROR: ResolutionScope is null \de4dot>pause			
Press any key to continue			

If instead you get a screen with some ERROR: lines, where one of them ends with Hmmmm... something didn't work - you used the wrong token.

2 Fixing "ResolutionScope is null"

- 1. In dnSpy, clear your workspace (File > Close All)
- 2. Then, do File > Open... and go to your EFT install location, then EscapeFromTarkov_Data/Managed/ and open ALL the files inside.

- 3. After that, do File > Open... once more, and go to wherever de4dot is located, and open Assembly-CSharp-cleaned.cll.
- 4. While the file is still selected in the "Assembly Explorer", do File > Save Module.... The "filename" field should have \Assembly-CSharp-cleaned-cleaned.dll at the end. Click Ok.

Save Module	×
Main MD Writer Options Metadata Cor20 PE	
Save PDB File Mixed-Mode Module	
✓ Shared Method Bodies	
Module Type DII 🔹	
Filename \de4dot\Assembly-CSharp-cleaned-cleaned.dll	
🛕 Run NGEN.exe if this assembly is installed in the GAC	
OK Cancel Res	set

That's it! You have a cleaned and deobfuscated assembly. Stay tuned for a guide on how to create .bpf patches, which can be used to update the assembly .bpf for the launcher with the assembly you just prepared.

3 Notes

If finding the token in the deobfuscation step fails, search manually through all ClassXXXX (**NOT GClassXXXX**) until you find a method that looks akin to this:

Code

```
// Token: 0x0600D56B RID: 54635 RVA: 0x0012870F File Offset: 0x0012690F
// Note: Class2056 might look different
public static string smethod_0(int int_0)
{
CurrentDomain.GetData(Class2056.string_0))[int_0];
}
```

Use the token of the found method and continue from there.

- Additional information -