

Forum:[Hack Forums](#)

Thread: [Adding resources to a compiled binary with Mono.Cecil](#)

Delirium™ - October 29, 2013, 1:37 pm

Heya C# folks.

I wrote this snippet to help people who are interested in Mono.Cecil or generally in C# learn how to add resources to a compiled binary or generally an existing .NET executable.

Of course, in order to use the following code, you need to have referenced Mono.Cecil.dll in your project and add `using Mono.Cecil;` on top of it.

The code is the following:

Add the following statements on top of your source file:

Code using Mono.Cecil;
`using System.Resources;`

Then within a class in the same source file, add the following:

```
Code public static string CreateResourceFile(byte[] FileData)
{
    string ResPath =
Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.Desktop), "res.resources");
    using (ResourceWriter Writer = new ResourceWriter(ResPath))
    {
        Writer.AddResource("Res", FileData);
        Writer.Generate();
    }
    return ResPath;
}
```

```
public static string CreateResourceFile(List<byte[]> Files)
{
    string ResPath =
Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.Desktop), "res.resources");
    using (ResourceWriter Writer = new ResourceWriter(ResPath))
    {
        int index = 0;
        foreach (byte[] ByteArray in Files)
        {
            Writer.AddResource(string.Format("Res{0}", index), ByteArray);
        }
        Writer.Generate();
    }
    return ResPath;
}
```

```
public static void InjectResourceFile(string TargetPE, string ResName, string ResourceFile)
```

```
{
    AssemblyDefinition ASMDef = AssemblyDefinition.ReadAssembly(TargetPE);
    EmbeddedResource EmbeddedRes = new EmbeddedResource(ResName + ".resources",
ManifestResourceAttributes.Public, File.ReadAllBytes(ResourceFile));
    ASMDef.MainModule.Resources.Add(EmbeddedRes);
    ASMDef.Write(TargetPE);
}
```

Example Usage:

```
Code string ResourceFile =
CreateResourceFile(File.ReadAllBytes("C:/Users/Coder/Desktop/ImageExample.jpg"));
InjectResourceFile("C:/Users/Coder/Desktop/TargetDotNetExe.exe", "Image", ResourceFile);
File.Delete(ResourceFile);
```

Deleting the generated resource file should be done in case you don't want it anymore or you're using the above code to inject resources on a crypter's (or generally any) stub. Also note, that I wrote an overload method which supports adding multiple resources! Using it should be easy, just follow the example above and instead of a byte[] parameter, pass a List<byte[]> parameter to it.

Enjoy.

Only3 - October 30, 2013, 1:49 pm

u can put it in 1

```
Code public static void AddResourceFile(string Target, byte[] FileData, string ResName)
{
    string ResPath =
IO.Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.Desktop),
"res.resources");
    using (Resources.ResourceWriter Writer = new Resources.ResourceWriter(ResPath)) {
        Writer.AddResource("Res", FileData);
        Writer.Generate();
    }
    AssemblyDefinition AsmDef = AssemblyDefinition.ReadAssembly(Target);
    EmbeddedResource EmbeddedRes = new EmbeddedResource(ResName +
Convert.ToString(".resources"), ManifestResourceAttributes.Public, File.ReadAllBytes(ResPath));
    AsmDef.MainModule.Resources.Add(EmbeddedRes);
    AsmDef.Write(Target);
    IO.File.Delete(ResPath);
}
```

Delirium™ - October 30, 2013, 3:05 pm

(Oct 30, 2013 01:49 PM)Only3 Wrote: [CITE](#) u can put it in 1

```
Code public static void AddResourceFile(string Target, byte[] FileData, string ResName)
{
    string ResPath =
IO.Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.Desktop),
"res.resources");
    using (Resources.ResourceWriter Writer = new Resources.ResourceWriter(ResPath)) {
        Writer.AddResource("Res", FileData);
        Writer.Generate();
    }
    AssemblyDefinition AsmDef = AssemblyDefinition.ReadAssembly(Target);
    EmbeddedResource EmbeddedRes = new EmbeddedResource(ResName +
Convert.ToString(".resources"), ManifestResourceAttributes.Public, File.ReadAllBytes(ResPath));
```

```
AsmDef.MainModule.Resources.Add(EmbeddedRes);  
AsmDef.Write(Target);  
IO.File.Delete(ResPath);  
}
```

I know, I just made the code cleaner.

.Death - October 31, 2013, 4:08 am

you can even use reflection to make it support obfuscated assemblies.
mono crashes quite often

Lugus - October 31, 2013, 11:26 am

Thank you for the tutorial mate. In fact I have a question for you and I apologize if it was a bit offensive.

I need to know if there is an actual use of this snippet, I mean why would you inject a random resource into a random binary unless the last one has a method that will use the injected resource and if that's the case, couldn't you just add the needed resources in the solution explorer it would a lot easier this way. Thank you for understanding.

Only3 - October 31, 2013, 11:37 am

u can use it in many ways if u can use ur brain

Delirium™ - October 31, 2013, 11:41 am

(Oct 31, 2013 11:26 AM)Lugus Wrote: [\[CITE\]](#) Thank you for the tutorial mate. In fact I have a question for you and I apologize if it was a bit offensive.

I need to know if there is an actual use of this snippet, I mean why would you inject a random resource into a random binary unless the last one has a method that will use the injected resource and if that's the case, couldn't you just add the needed resources in the solution explorer it would a lot easier this way. Thank you for understanding.

You can use it in crypters if you're deploying them with a precompiled stub. You can use it in other forms of programs that require stubs with variable resources.