

Forum:[Hack Forums](#)

Thread: [Creating a launcher..](#)

Alien - March 23, 2020, 9:19 pm

I've gone back into an old Launcher project from a few years ago and have decided I'm gonna make it into a fully working project. Basically, the program reads an executable's byte code and runs it in memory. This part is already taken care of and I already have working code for this function. Now what I want to do is, be able to load the .exe via an OpenFileDialog and generate a new executable file that has the loaded .exe as a my.resource within it so that I can drop it without it ever being detected by the user.

I am aware of CodeDom and believe it is most likely the best option to go with, but if anyone has any better ideas/suggestions I am open to them. Thanks. I want to do this the best way possible and to ultimately learn from this project.

Accord - March 23, 2020, 10:07 pm

Codedom, mono.cecil, and dnlib is what ive used for creating new assembly.

Alien - March 23, 2020, 10:35 pm

(Mar 23, 2020 10:07 PM)SIM-4463207 Wrote: [\[CITE\]](#) Codedom, mono.cecil, and dnlib is what ive used for creating new assembly.

I'll check into Mono and dnlib. Thanks!

I'm currently in the process of getting CodeDom working. So far I've gotten it to add my executable as a resource file and generate a new .exe. Next task is to be able to execute it in the newly generated exe file from the my.resources.

StarZ - March 24, 2020, 2:45 pm

(Mar 23, 2020 09:19 PM)Alien Wrote: [\[CITE\]](#) I've gone back into an old Launcher project from a few years ago and have decided I'm gonna make it into a fully working project. Basically, the program reads an executable's byte code and runs it in memory. This part is already taken care of and I already have working code for this function. Now what I want to do is, be able to load the .exe via an OpenFileDialog and generate a new executable file that has the loaded .exe as a my.resource within it so that I can drop it without it ever being detected by the user.

I am aware of CodeDom and believe it is most likely the best option to go with, but if anyone has any better ideas/suggestions I am open to them. Thanks. I want to do this the best way possible and to ultimately learn from this project.

My suggestion is not to drop anything. You can define an assembly and set it's content to execute it via a reflection runpe.

Edit:

Code ' thanks to Deathader for the reflection runpe...

```
Private Sub RunFromMemory(ByVal bytes As Byte()) 'not sure how it works :(
    Dim assembly As Assembly = assembly.Load(bytes)
    Dim entryPoint As MethodInfo = [assembly].EntryPoint
    Dim objectValue As Object =
RuntimeHelpers.GetObjectValue([assembly].CreateInstance(entryPoint.Name))
```

```
entryPoint.Invoke(RuntimeHelpers.GetObjectValue(objectValue), New Object() {New String() {"1"}})
```

End Sub

Yoinked from <https://hackforums.net/showthread.php?tid=4391216>

May have to play around with it to drop detections. But, It works.

BahNahNah - March 24, 2020, 9:52 pm

(Mar 24, 2020 02:45 PM)StarZ Wrote: [\[CITE\]](#) My suggestion is not to drop anything. You can define an assembly and set it's content to execute it via a reflection runpe.

Edit:

Code ' thanks to Deathader for the reflection runpe...

```
Private Sub RunFromMemory(ByVal bytes As Byte()) 'not sure how it works :(
```

```
Dim assembly As Assembly = assembly.Load(bytes)
```

```
Dim entryPoint As MethodInfo = [assembly].EntryPoint
```

```
Dim objectValue As Object =
```

```
RuntimeHelpers.GetObjectValue([assembly].CreateInstance(entryPoint.Name))
```

```
entryPoint.Invoke(RuntimeHelpers.GetObjectValue(objectValue), New Object() {New String() {"1"}})
```

End Sub

Yoinked from <https://hackforums.net/showthread.php?tid=4391216>

May have to play around with it to drop detections. But, It works.

What even is that code yikes.

Code Dim objectValue As Object =

RuntimeHelpers.GetObjectValue([assembly].CreateInstance(entryPoint.Name))This is going to return null. CreateInstance is going to attempt to create an instance of the type "Main", wich is not going to exist.

You can simplify it to just

Code Dim asm As Assembly = Assembly.Load(bytes)

asm.EntryPoint.Invoke(null, new string({})

Alien - March 24, 2020, 10:16 pm

A little update: I've gotten a lot closer to the end goal.

Things that have been completed:

1. it Allows me to import a .exe file as a resource via openFileDialog and uses Codedom to generate the new Launcher.exe program with the .exe hidden inside the resources.

2. The newly generated Launcher.exe has a Launch button within it. Once the button is clicked it drops the .exe from its my.resources into the current directory that the launcher is running from.

3. I have working code that runs it from memory but i'm currently not at the point where its needed or can be used the way I want it to be used just yet.

The steps still needed to be completed:

1. I need to find a way to have better access to the file.. I thought that dropping it for a brief second,

reading it as bytes and then putting it into a variable for later usage (running it through memory), and deleting it, was the best plan. But at this point it seems to chaotic and it may possibly get detected by AV's in the future. Even so, for now this is what I am currently working at unless someone gives me a better idea.

My goal overall:

it needs to be as silent as possible meaning I don't want it to be dropping 10 million files etc. I'd rather most of the work to be done internally within the program if possible.

StarZ - March 24, 2020, 11:19 pm

(Mar 24, 2020 09:52 PM)BahNahNah Wrote: [\[CITE\]](#) What even is that code yikes.

`[/code]`

Old code yinked from google. Mainly to show a basic example of the concept in question. Thank you for posting a better solution. ?

(Mar 24, 2020 10:16 PM)Alien Wrote: [\[CITE\]](#) The steps still needed to be completed:

1. I need to find a way to have better access to the file.. I thought that dropping it for a brief second, reading it as bytes and then putting it into a variable for later usage (running it through memory), and deleting it, was the best plan. But at this point it seems to chaotic and it may possibly get detected by AV's in the future. Even so, for now this is what I am currently working at unless someone gives me a better idea.

My goal overall:

it needs to be as silent as possible meaning I don't want it to be dropping 10 million files etc. I'd rather most of the work to be done internally within the program if possible.

Take a look at [this](#) old project on github. It shows a lazy way to store the payload. It's far from the best. However it is a viable solution for small payloads.

Alien - March 25, 2020, 12:04 am

(Mar 24, 2020 11:19 PM)StarZ Wrote: [\[CITE\]](#) Old code yinked from google. Mainly to show a basic example of the concept in question. Thank you for posting a better solution. ?

Take a look at [this](#) old project on github. It shows a lazy way to store the payload. It's far from the best. However it is a viable solution for small payloads.

Going to take a look at it soon. Thanks! ?

i'll update everyone on the progress and where I'm at with the project tomorrow.

Update:

After getting a good night sleep I woke up and immediately got back on the project. The program is now fully functional and does everything I wanted/needed it to do. I will add this to Github soon and make a separate thread for this project so other's can learn from the code as well.