

Forum: [Hack Forums](#)

Thread: [Dot Net Assembly Jail | Trying to expose the intent](#)

Barret - September 1, 2017, 1:00 pm

For a confined environment to execute an assembly with custom set of permissions that you control.

Not all assembly types work entirely 100% within a boxed environment but most of them will expose their intent within this test environment as shown above with the Jail.Test i supplied.

You have a big list of all sorts of permissions, you can find them here:
[https://msdn.microsoft.com/en-us/library/system.security.permissions\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.security.permissions(v=vs.110).aspx)

And you can add them like so:

PHP Code

```
<?php
```

```
confinement.Permissions.AddPermission(New SecurityPermission(SecurityPermissionFlag.
Execution))
```

```
confinement.Permissions.AddPermission(New FileIOPermission(FileIOPermissionAccess.
AllAccess, "C:\SomeFolder\"))
```

The Drag&Drop feature is just for Console or Window assemblies, i supplied an additional Jail.Test project which sole purpose is to perform debugging and testing of certain permissions.

To perform a jailed exeuction, drag the Jail.Test.exe onto the jail image of the Jailer application itself and the log will tell you whats going on.

If an assembly has DLL's, you can drag them along with it and they will be loaded into that jailed domain as well.

DLL's have two different approaches, DynamicProxy and TypedProxy.

Dynamic Proxy

Parameters: String[Namespace.Class]

Returns: Object

PHP Code

```
<?php
```

```
jail.ToDynamicProxy("Namespace.Class").MyMethod()
```

TypedProxy

Parameters: (Of Contract Type), String[Namespace.Class]

Returns: Defined by contract

PHP Code

```
<?php
```

```
jail.ToTypedProxy(Of Contract)("Namespace.Class").MyMethod()
```

Remark:

A typed proxy is made by a contract class that makes up the interface of the jailed assembly, the methods and properties are assumed to be the same in signature and name reference.

Quick demonstration (a test project is supplied within the github link):

PHP Code

```
<?php
```

```
Private Sub JailCreateAndRun(Assemblies() As String)
```

```
    '// Collect and append assemblies into our collection
```

```
    Dim Collection As New Collection
```

```
    For Each filename As String In Assemblies
```

```
        If (IO.File.Exists(filename)) Then
```

```
            Collection.Load(New Loaders.ByFilename(filename))
```

```
        End If
```

```
    Next
```

```
    '// Create confinement
```

```
    Using confinement As New Confinement("My Jail")
```

```
        '// Bare minimum (Execution and local path access)
```

```
        confinement.Permissions.AddPermission(New SecurityPermission(SecurityPermissionFlag.  
Execution))
```

```
        confinement.Permissions.AddPermission(New FileIOPermission(FileIOPermissionAccess.All  
Access, confinement.Path))
```

```
        '// GUI permission
```

```
        confinement.Permissions.AddPermission(New UIPermission(UIPermissionWindow.  
AllWindows))
```

```
    '// Create jail with confinement and collection, will call back for load results
```

```
    Using jail As Jail = CType(jail.Create(confinement, AddressOf Me.LoadCallback, Collection),  
Jail)
```

```
        Try
```

```
            '// Assign event handler
```

```
            AddHandler jail.JailBlockedPermission, AddressOf Me.JailBlockedPermission
```

```
            '// Run application (assuming it has an entrypoint)
```

```
            jail.Run(New Object() {New String() {}})
```

```
        Finally
```

```
            '// Remove event handler
```

```
        RemoveHandler jail.JailBlockedPermission, AddressOf Me.JailBlockedPermission
    End Try
End Using
End Using
End Sub
```

```
// Callback result
Private Sub LoadCallback(sender As Object, loaded As Boolean, ex As Exception)
    If (Not loaded) Then
        '....Something wend wrong while loading an assembly...
    End If
End Sub
```

```
// Callback blocked security events
Private Sub JailBlockedPermission(Sender As Object, Request As Request)
    '...a permission demand was made by the assembly but denied, info is stored in Request
End Sub
```

The project is just at a test stage, bugs are to be expected, leave a comment for questions or fork/download the project.

Github: <https://github.com/TheBarret/DotNetJail>

- B.

DisguiserNET - September 2, 2017, 4:39 am

this is hf, people cant fud their crypter with that, so they wont like it.
must be the reason why a high quality thread hasnt received any response yet

Barret - September 2, 2017, 8:59 am

(Sep 2, 2017 04:39 AM)DisguiserNET Wrote: [\[CITE\]](#) this is hf, people cant fud their crypter with that, so they wont like it.
must be the reason why a high quality thread hasnt received any response yet
Hahaha, the Good and the Bad are never far apart from eachother. ?

DBots - September 2, 2017, 10:46 am

This reminds me of Sandboxie.

I could see myself using it in Windows' context menu as Open With -> Jail.

Have fun with this awesome project!

Barret - September 2, 2017, 12:44 pm

(Sep 2, 2017 11:35 AM)Orka Wrote: [\[CITE\]](#) Very cool. How does the settings contract work?
Its much like dynamic method callings, you can setup a class that inherits MarshalByRefObject and then you cast the proxy
into that contract class, so you can call the methods, the proxy will then try exchange the invoke to the underlying assembly aka jailed assembly

These approaches are available as of right now:

Indirect entrypoint execution (first assembly that has one will be invoked)
Code Jail.Run(New Object() {})
Direct entrypoint execution (same as above but now with specified type)
Code Jail.Run("Untrusted.Assembly", New Object() {})
Dynamic method or Typed (contract) method invoke
Code Jail.ToDynamicProxy("SomeType").Method()
Jail.ToTypedProxy(Of Contract).Method()

Example:

Barret - September 3, 2017, 11:09 am

I managed to create Drag & Dropper to execute a assembly, did some testing with accessing windows folder or register values. all perfect, ill soon be releasing this.

cawkre - September 3, 2017, 11:12 am

This looks amazing!
nice work!

Barret - September 3, 2017, 8:17 pm

Alright, source is released, have fun playing :)

Tsukiri - September 9, 2017, 4:09 am

Looks like a good quality project, thanks for sharing it.
The UI is horrible though ;*

Ninja - September 9, 2017, 5:29 am

(Sep 9, 2017 04:09 AM)Tsukiri Wrote: [\[CITE\]](#) Looks like a good quality project, thanks for sharing it.
The UI is horrible though ;*
The classic Windows theme is a godsend. Wash your mouth!

Effexion - September 9, 2017, 5:44 am

(Sep 9, 2017 04:09 AM)Tsukiri Wrote: [\[CITE\]](#) Looks like a good quality project, thanks for sharing it.
The UI is horrible though ;*
OP obviously wanted to focus on actual functionality over a pointlessly polished GUI. The source is there so no one is forced to use it.

Ninja - September 9, 2017, 8:04 am

(Sep 9, 2017 05:44 AM)Effexion Wrote: [\[CITE\]](#) OP obviously wanted to focus on actual functionality over a pointlessly polished GUI. The source is there so no one is forced to use it.
The GUI looks fine imo.
Why people hating on the classic UI? I think it looks fine as fuck.

Tsukiri - September 10, 2017, 2:20 am

(Sep 9, 2017 08:04 AM)Ninja Wrote: [\[CITE\]](#) The GUI looks fine imo.
Why people hating on the classic UI? I think it looks fine as fuck.
I think it looks very cheap and unprofessional. Obviously I get why OP doesnt care about the GUI too much, ofcourse functionality is way more important. But cant lie about the fact that in my opinion it looks very ugly :p

Ninja - September 10, 2017, 5:38 am

(Sep 10, 2017 02:20 AM)Tsukiri Wrote: [\[CITE\]](#) I think it looks very cheap and unprofessional. Obviously I get why OP doesn't care about the GUI too much, ofcourse functionality is way more important. But can't lie about the fact that in my opinion it looks very ugly :p
You got any idea what the software that the military (in my country at least) looks like? It doesn't look flashing, but under the hood it (I assume) is.

Tsukiri - September 10, 2017, 12:26 pm

(Sep 10, 2017 05:38 AM)Ninja Wrote: [\[CITE\]](#) You got any idea what the software that the military (in my country at least) looks like? It doesn't look flashing, but under the hood it (I assume) is.
No idea, but ofcourse, like I said, functionality is always number one.
However, it doesn't take a lot of work to add a basic GUI that doesn't look like a complete disaster, hehe.

However it's opensource like someone mentioned, so it doesn't matter. Was just wondering why people code such great tools and then make it look like shit :p

Barret - September 10, 2017, 12:46 pm

(Sep 10, 2017 12:26 PM)Tsukiri Wrote: [\[CITE\]](#) No idea, but ofcourse, like I said, functionality is always number one.
However, it doesn't take a lot of work to add a basic GUI that doesn't look like a complete disaster, hehe.

However it's opensource like someone mentioned, so it doesn't matter. Was just wondering why people code such great tools and then make it look like shit :p
Simple answer lad, its not relevant, its pure for demonstration sake. I have a question for you now, why are we arguing over something that isn't relevant?

Tsukiri - September 10, 2017, 1:05 pm

(Sep 10, 2017 12:46 PM)Barret Wrote: [\[CITE\]](#) Simple answer lad, its not relevant, its pure for demonstration sake. I have a question for you now, why are we arguing over something that isn't relevant?
I am not arguing about anything, I was just wondering. Try re-reading what I said. I already mentioned multiple times that it is obvious that functionality > looks.