

Forum:[Hack Forums](#)

Thread: [How to dynamically get c# API functions?](#)

Tr0jandog - December 20, 2020, 4:23 am

Hello yall

so as you you all know, in C we can simply use `loadlibrary+getprocaddr` to get the address, or even parse the TEB/PEB to find the addresses without even using a single API

but, how to do it for C# APIs?

for example, in my crypter i am using these APIs :

```
Assembly.GetExecutingAssembly()  
ResourceManager()  
CreateDecryptor()  
CryptoStream()  
..
```

now these are often could lead to detection, how can i find them and use them just like in C, without directly using them?

note that I'm asking about .NET APIs, I'm not looking for a way to call `kernel32/ntdll` APIs in c# as I already know how.

any open source project that demonstrates this in c#?

riu41776 - December 20, 2020, 5:39 am

lol just `DllImport LoadLibrary` and `GetProcAddress`.

Ok, so you're talking about a managed code dll, i know how to do it statically by referencing the dll/class but dynamically you can use `Reflection`.

HFDone - December 20, 2020, 5:41 am

same as in C use APIs

search the section for dynamic api call

Tr0jandog - December 20, 2020, 7:29 am

(Dec 20, 2020 06:39 AM)riu41776 Wrote: [\[CITE\]](#) lol just `DllImport LoadLibrary` and `GetProcAddress`.

Ok, so you're talking about a managed code dll, i know how to do it statically by referencing the dll/class but dynamically you can use `Reflection`.

(Dec 20, 2020 06:41 AM)HFDone Wrote: [\[CITE\]](#) same as in C use APIs
search the section for dynamic api call

But as i said i want to call .NET apis like `Assembly.GetExecutingAssembly()`, i dont want to call functions inside `ntdll/kernel32`

which dll should i import using `dllimport/loadlibrary` and which function inside of it should i call to dynamically call `GetExecutingAssembly`?

as far as i know .NET executables only import `mscorlib.dll` with only `CorExeMain` function

There has to be some structure like PEB in .net executables that i can use to find .net APIs and call them directly

i know that library calls in .NET are in a way that they reference an index inside a table, but not sure how can i use this to call the API in runtime

HFDone - December 20, 2020, 7:47 am

Load the Dll `Assembly.Load` (or `Assembly.LoadFrom`)

Call `Assembly.GetTypes` for each type call `Type.GetMethods` to get the Method
Get `MethodInfo` and call `MethodInfo.Invoke`

TITANIUM_DZ - January 16, 2021, 9:53 pm

you can use `Method CallByName` From reference `Microsoft.VisualBasic`

```
Interaction.CallByName(  
    AppDomain.CurrentDomain,  
    "Load",  
    CallType.Method,  
    Convert.FromBase64String("base64")  
);
```

gigajew - January 17, 2021, 8:29 am

<https://github.com/gigajew/CallByString> for regular .NET stuff. Stay tuned to the git.

SIM-4717218 - January 18, 2021, 7:44 am

look at <http://pinvoke.net/>