Background

All of the C# code in this repository is the work of Nick Tyrer, I have simply compiled each of the examples for .Net 2 (Win7?) and .Net 4 (Win10?) with some updated libraries.

The Net 4.0 binaries are compiled against .Net 4.5 as that it the version used by the PowerShell library referenced.

The original code used the UnmanagedExports library which is used to binary patch the output files to allow them to provide native exports e.g. like a standard C DLL. I

I have updated the .Net 4.0 versions to use the DllExport library which underneath uses the UnmanagedExports library but appears to be have been updated. The was also the factor that I could not get the UnmanagedExports library to work with VS2017.

Nick Tyrer's code was originally identified from the blog of SubTee, which if you are interested in application blocking bypasses then it should be your first call.

Bypasses

The follows lists the three bypasses that have been compiled. All of the bypasses provide an interactive PowerShell shell.

cpl

The **Control Panel (cpl)** bypass was from the following GitHub Gist by Nick Tyrer:

https://gist.github.com/NickTyrer/b1b9694d909352a9a72d9b82319ca8f4

To run the bypass execute the following command. Note the the path to the DLL must be fully qualified:

```
control .\c.dll
```

odbcconf

The following link provides background on the **odbcconf** bypass:

https://subt0x10.blogspot.co.uk/2016/10/command-line-camouflage-odbcconfexe.html

The **odbcconf** example comes from the following Github Gist by Nick Tyrer:

https://gist.github.com/NickTyrer/6ef02ce3fd623483137b45f65017352b

To run the bypass execute the following command, ensuring that the **file.rst** is in the same location and the DLL file is correctly identified in the file. Note that the DLL cannot be called **odbcconf.dll** as it won't execute!:

```
odbcconf -f file.rsp
```

msiexec

The **msiexec** example comes from the following Github Gist by Nick Tyrer:

https://gist.github.com/NickTyrer/9f8cbd5142c4cea63e98da8aac39c874b

To run the bypass execute the following command. Note the the path to the DLL must be fully qualified:

```
msiexec .\m.dll
```

Links

- https://twitter.com/nicktyrer
- https://subt0x10.blogspot.co.uk
- https://github.com/subTee
- https://github.com/3F/DllExport
- https://sites.google.com/site/robertgiesecke/Home/uploads/unmanagedexports
- https://www.nuget.org/packages/DllExport/
- https://www.nuget.org/packages/UnmanagedExports