针对马拉维(Malawi)国民银行的网络攻击样本分析报告

安天安全研究与应急处理中心（Antiy CERT）

初稿完成时间：2018 年 11 月 22 日 15 时 10 分

首次发布时间：yyyy 年 mm 月 dd 日 hh 时 mm 分

本版更新时间：2018 年 11 月 26 日 21 时 02 分

扫二维码获取最新版报告

目 录

1 概述

近日，安天 CERT（安全研究与应急处理中心）在梳理相关安全事件时发现多例对马拉维国民银行（National Bank of Malawi）的钓鱼邮件攻击样本。马拉维共和国（Republic of Malawi）是位于非洲东南部的内陆国家，国土面积 11.8 万平方公里。截至 2017 年 9 月，其全国共计人口 2840 万人，是非洲撒哈拉沙漠以南人口密度最高的国家之一。其国家经济长期依赖农业出口，政府常年靠举债度日，曾被联合国评为世界上最贫困的国家之一（摘自维基百科），国民银行是其国内最大的商业银行。

安天通过对攻击样本和关联线索的综合分析，发现这是一系列以金融机构电子邮件为突破口，通过邮件进行交叉渗透的定向攻击事件。攻击者使用对陈旧漏洞的免杀技巧构造攻击载荷执行入口，而攻击载荷采取对多个开源代码进行改写和重新编译，将二进制远控木马加密嵌入其中，实现内存执行，达到免杀和绕过安全机制的效果。

在这一系列攻击事件中，有四家马拉维国民银行的地方分行，成为攻击者的重要目标，其中大南部区（southend）官方客服邮箱已经被攻击者盗用。攻击者利用事先从国民银行部分地区分行盗取的官方邮箱口令，向其他分行工作人员发送带有恶意文档附件的邮件，作为附件的恶意文档利用 CVE-2014-6352 漏洞[1]发起攻击。此漏洞可以绕过“沙虫”漏洞（SandWorm）补丁 MS14-060 的安全保护。漏洞利用成功后，样本会执行名为“Target.scr”的可执行程序，该程序由攻击者基于开源代码[2]修改编译生成，攻击者在重写了 main 函数代码，程序运行时并不会调用开源代码原有的功能函数，而是内存展开执行内嵌的 DarkComet 远控木马，进而向目标系统发起攻击。

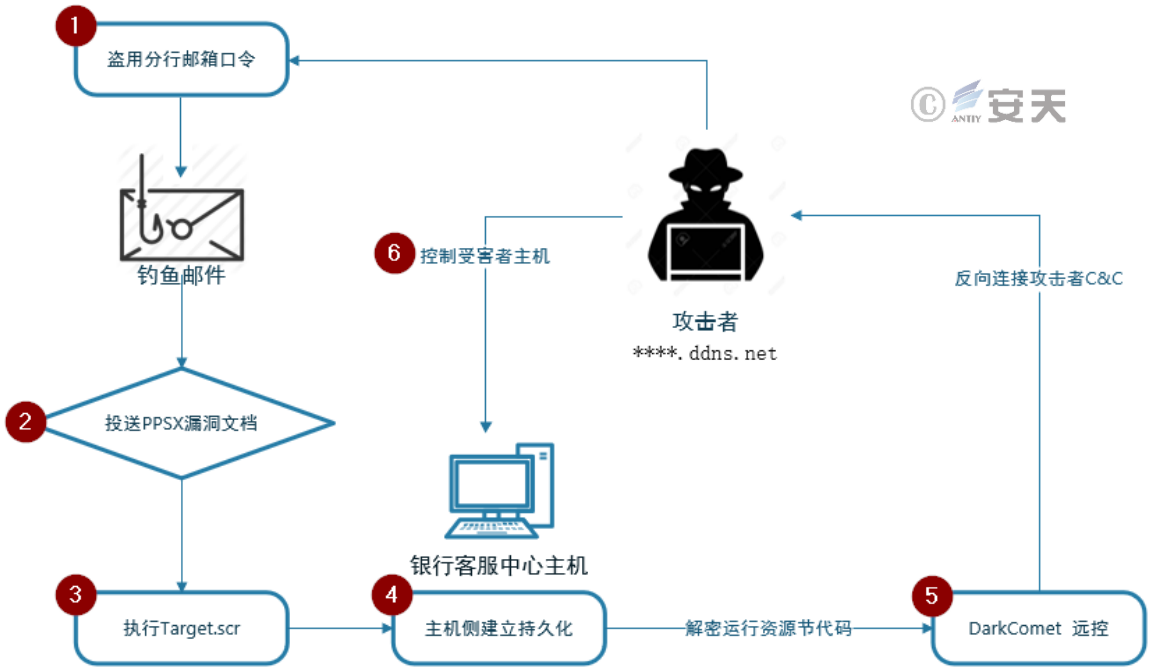


图 1-1 攻击流程示意图

这一系列邮件均来自一台固定的主机和相对固定的 IP 网段。通过对攻击者活跃的网段进行检索，我们发现其所涉及的两个 IP（185.82.***.***和 185.82.***.***），物理位置显示位于新西兰，曾于今年 10 月初绑定过一系列动态域名厂商提供的临时二级域名，并被作为远控木马的回传 C&C 服务器；而同一网段的 IP 185.82.***.***则于今年 10 月初被用于挂载知名勒索软件 GandCrab。

1.1 钓鱼邮件详情

钓鱼邮件<1>于马拉维时间 2018 年 10 月 4 日（周四）上午 10:43 被发送到马拉维国民银行大南部区（southend）客服中心某员工所管理的官方邮箱<southend@natbankmw.com>中（可能由于地理位置紧邻和管理成本问题，大南部区（southend）客服中心并入了利隆圭（Lilongwe）地区分行）。发件人为“Yousef Syda”，是马拉维国民银行萨利马（Salima）分行的员工，初步判断该邮箱已被盗。

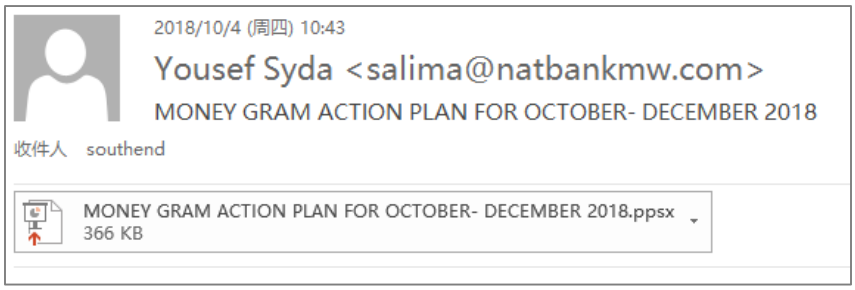


图 1-2 钓鱼邮件<1>内容

钓鱼邮件<2>于马拉维时间 2018 年 10 月 4 日（周四）上午 10:52 被发送到马拉维国民银行松巴（Zomba）分行经理 Thomas Chimkowola 所管理的官方邮箱<zomba@natbankmw.com>中，显示发件人仍为“Yousef Syda”，时间上晚于钓鱼邮件<1>约 9 分钟。

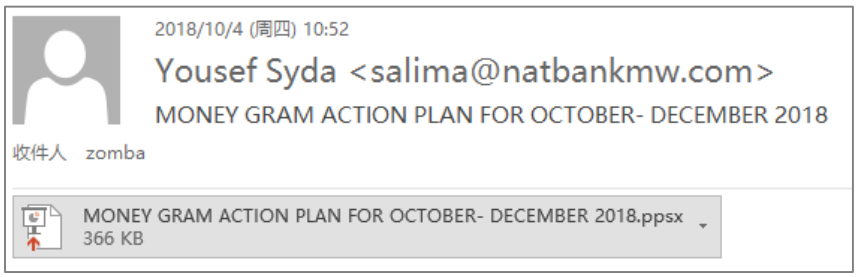


图 1-3 钓鱼邮件<2>内容

被盗邮箱 Yousef Syda <salima@natbankmw.com>两次发信的 IP 地址均为 185.82.***.***。

```
Received: from MAIL.natbankmw.com (96.1.200.180) by mail.natbankmw.com
 (96.1.200.180) with Microsoft SMTP Server (TLS) id 15.0.1178.4; Wed, 3 Oct
 2018 21:50:44 +0200
Received: from [redacted] (185.82.[redacted]) by MAIL.natbankmw.com
 (192.168.0.4) with Microsoft SMTP Server id 15.0.1178.4 via Frontend
 Transport; Wed, 3 Oct 2018 21:50:44 +0200
Return-Path: salima@natbankmw.com
```

图 1-4 被盗邮箱的 IP 地址

两份邮件带有同一标题“MONEY GRAM ACTION PLAN FOR OCTOBER- DECEMBER 2018（2018 年 10 月到 12 月 MoneyGram 汇款计划）”，无正文，邮件标题字元集属性为 UTF-8，并附有同一恶意文档“MONEY GRAM ACTION PLAN FOR OCTOBER- DECEMBER 2018.ppsx”，该文档使用漏洞 CVE-2014-6352[0]进行攻击，该漏洞可绕过“沙虫”漏洞（SandWorm）补丁 MS14-060 的安全保护。在接收人双击打开 ppsx 文档后，其会自动开始 Slide Show 播放模式，触发 Verb 动作，执行临时目录下释放的恶意代码“Target.scr”。

1.2 被攻击者信息

受害者的邮箱域名 natbankmw.com 归马拉维国民银行所有[3]。马拉维国民银行成立于 1971 年，由起源于南非的标准银行（Standard Bank）和英国巴克莱银行 DCO（Dominion Colonial Overseas）合并成立，现共拥有 850 名员工，总部坐落于马拉维的金融和商业中心布兰太尔市（Blantyre）。截至 2016 年 12 月，该银行的总资产价值为 3140 亿 MWK（马拉维克瓦查，约合 4.4 亿美元），是马拉维最大的商业银行[4]。

图 1-5 马拉维国民银行总部建筑（纬度：-15.8 经度：35.0）

截至 2018 年 4 月，马拉维国民银行在马拉维全国维持了 30 多家分行，简称“客服中心”（Service Centers）。但由于银行缺乏资金和运营问题，多地“客服中心”管理混乱。2016 年曾经发生巴拉卡（Balaka）地区分行的在职员工监守自盗取走上百万 MWK 银行存款的事件[5]。此次网络攻击事件也正是首先针对地区分行的薄弱点，盗取分行邮箱向其他分行员工发送钓鱼邮件的典型案例。

图 1-6 此次攻击事件受害者之一，马拉维国民银行利隆圭（Lilongwe）分行

图 1-7 此次攻击事件受害者之一，马拉维国民银行松巴（Zomba）分行

图 1-8 被盗邮箱所属分行，马拉维国民银行萨利马（Salima）分行

事件相关的三个邮箱皆由国民银行三个地区分行“Salima Malawi”、“Zomba Malawi”及“Lilongwe Malawi”的员工所管理，邮箱地址均可从银行官方网站客服主页上获取，被盗取的是第一位员工 Yousef Syda <salima@natbankmw.com>的邮箱，可能是攻击者完成的第一个突破口。

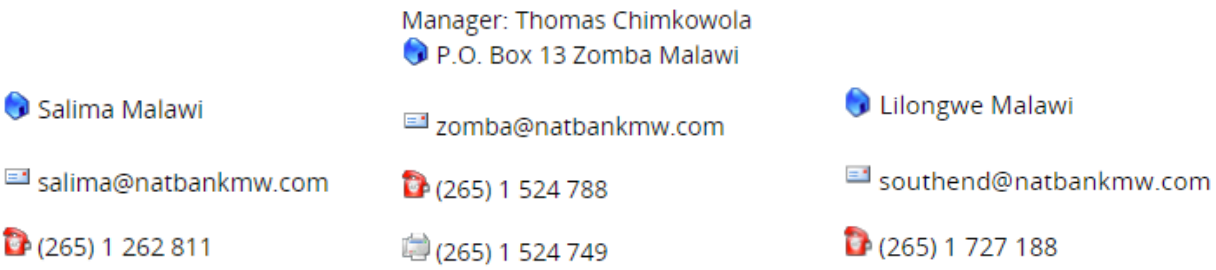


图 1-9 公开在国民银行官网上的地区客服信息

2 样本分析

2.1 PPSX 样本分析

样本“MONEY GRAM ACTION PLAN FOR OCTOBER- DECEMBER 2018.ppsx”由钓鱼邮件投递，利用了著名的“沙虫”（SandWorm）漏洞补丁的绕过漏洞 CVE-2014-6352[0] 。沙虫漏洞是 OLE 包管理 INF 任意代码执行漏洞，存在于 Microsoft Windows 服务器上的 OLE 包管理器，该漏洞影响 Win Vista、Win7 等以上操作系统，攻击者使用 PowerPoint 作为攻击载体。通过利用该漏洞可以在 OLE 打包文件（packer.dll）中下载并执行类似的 INF 外部文件，从而达到攻击者执行任意命令的目的。微软为沙虫漏洞发布了 MS14-060 补丁，但通过构造特殊的 CLSID 和 Verb 动作，该补丁可以被漏洞 CVE-2014-6352 绕过。之后微软再次发布 MS14-064 补丁针对 CVE-2014-6352 进行了修补。

表 2-1 事件中使用的 ppsx 样本标签

病毒名称	**Trojan[Exploit]/OLE.CVE-2014-6352**
原始文件名	MONEY GRAM ACTION PLAN FOR OCTOBER- DECEMBER 2018.ppsx
MD5	96EFB1EF94363045329C1DDB606AAD1F
文件大小	365.4 KB (374,199 bytes)
文件格式	Document/Microsoft.PPTX[:PowerPoint 2007-2013]
最后修改用户	Windows User
文档创建时间	2018-10-03T16:09:12Z
最后一次修改	2018-10-03T16:10:34Z
VT 首次上传时间	2018-10-05 01:20:47 UTC
VT 检测结果	21 / 60

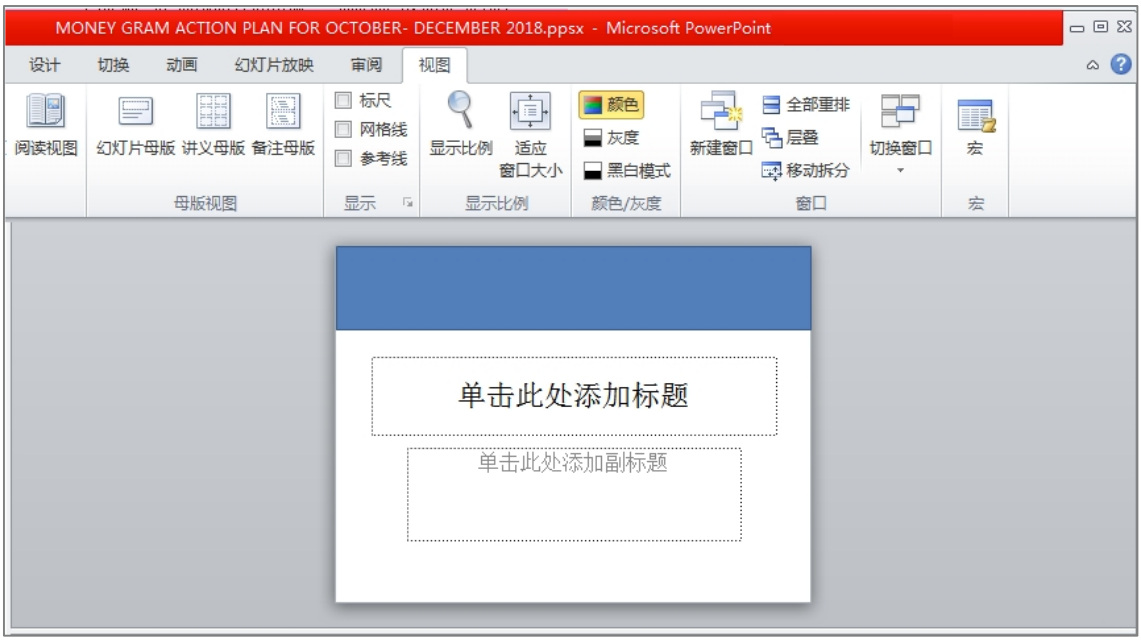


图 2-1 打开后的 PPSX 漏洞攻击文档内容为空

PPSX 相关文件内容为空，没有进行相关的内容构造和伪装。解压 PPSX 文档可以找到“\ppt\slides\slide1.xml”指定了嵌入 OLE 对象的 ID 为 rId3。

```
<p:oleObj spid="_x0000_s1026" name="Packager Shell Object"
showAsIcon="1" r:id="rId3" imgW="529200" imgH="437760" progId="Package"><p:embed/></p:oleObj>
```

文档“\ppt\slides_rels\ slide1.xml.rels”指定了 rId3 对应目录“\ppt\embeddings\”下的 oleObject1.bin

```
<Relationship Id="rId3" Type="http://schemas...ips/oleObject" Target="../embeddings/oleObject1.bin"/>
```

文件“\ppt\embeddings\oleObject1.bin”为嵌入了 PE 可执行程序的 OLE Package 对象。

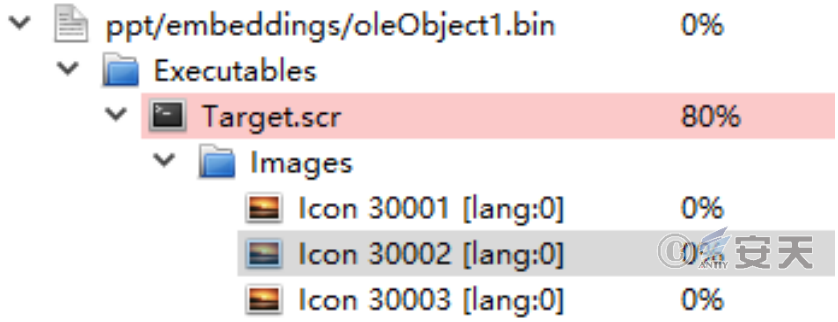


图 2-2 样本嵌入了包含 PE 的 OLE 对象

OLE Package 对象的“Olen0Native”流包含了一个完整的 PE 文件，流开头字符串能看出 PE 文件在攻击者机器上的原始保存路径“C:\Users\Analiz\Desktop\Target.scr”以及释放的路径“%TEMP%\Target.scr”。

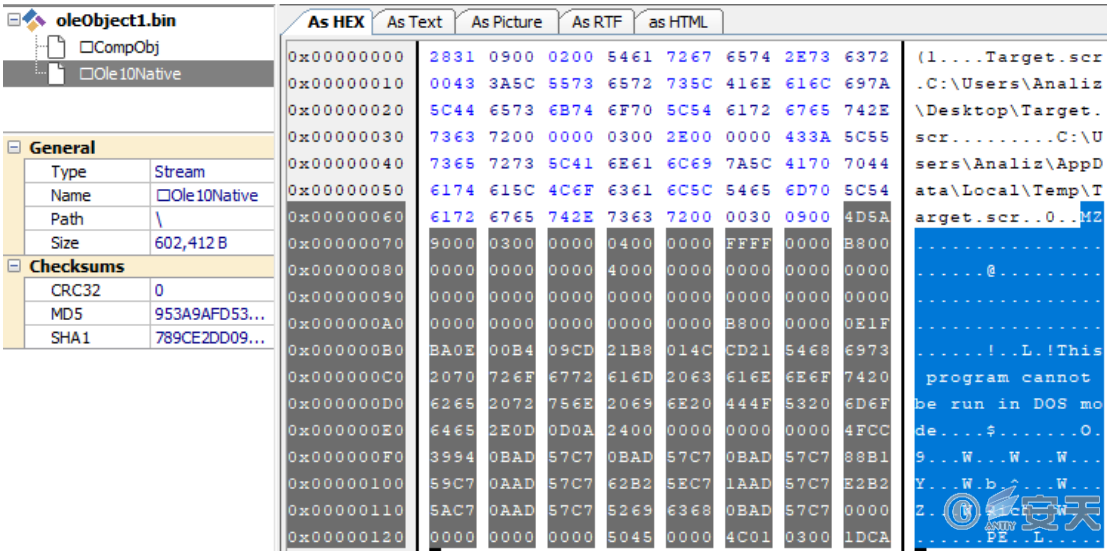
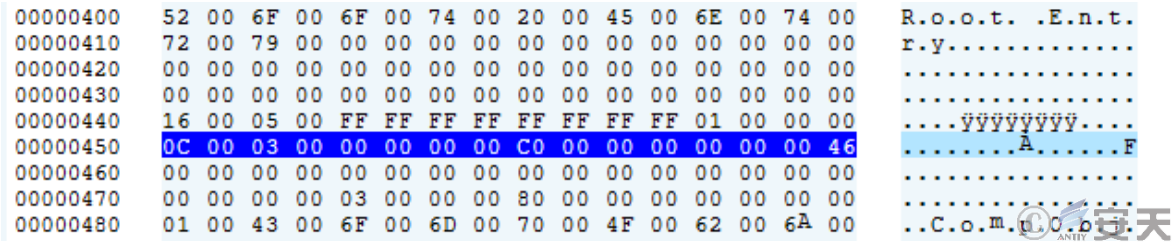


图 2-3 OLE 对象包含了完整的 PE 数据“Target.scr”和原始保存路径信息

模块 packager.dll 调用 CPackage::Load 读入该 OLE Package 对象时，首先调用 ReadClassStg 获取 OLE 复合文档的 CLSID 判断文件类型，该样本 OLE 对象的 CLSID：{0003000c-0000-0000-00c0-000000000046}。



CLSID 对应 OldPackage 类型，因此之后依次调用 CPackage::PackageReadFromStream →CPackage::EmbedReadFromStream→ CopyStreamToFile 将流数据拷贝至临时目录具体文件。

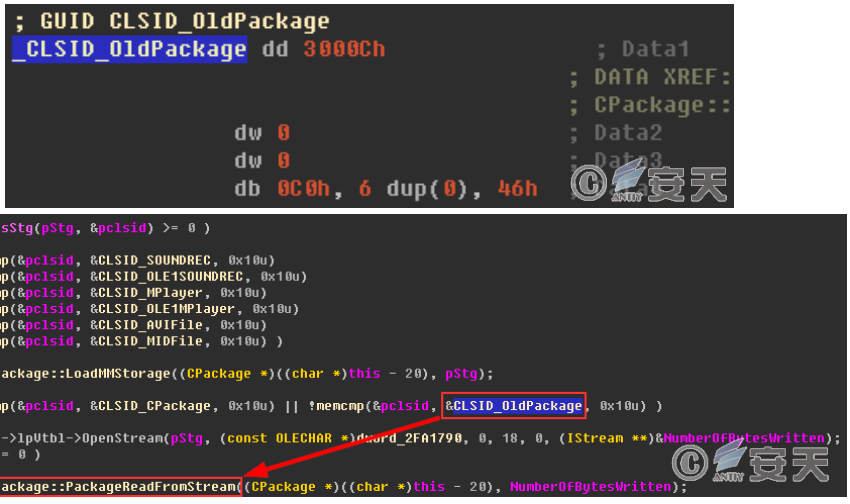


图 2-4 OLE 对象被判断为 OldPackage 类型，流数据被读入

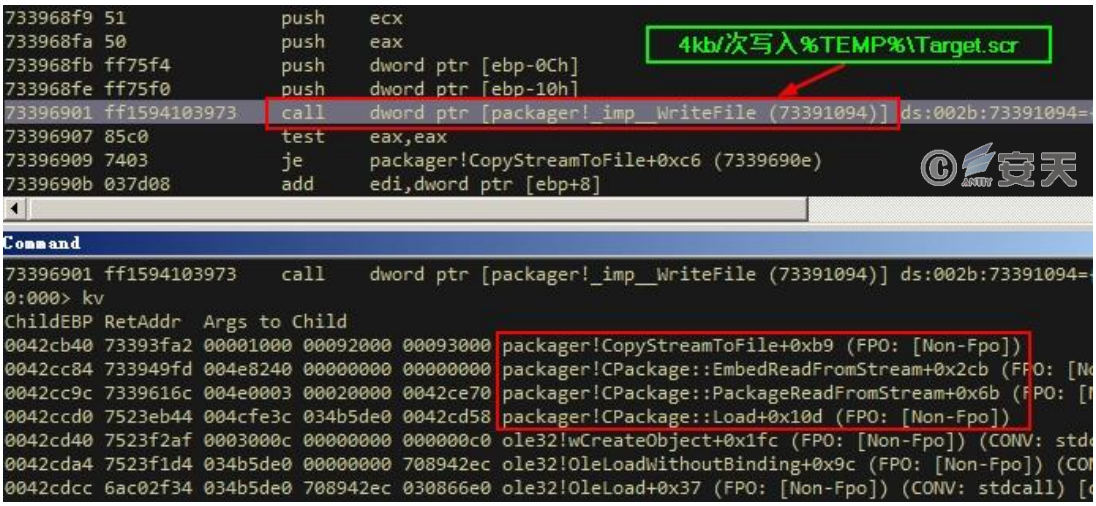


图 2-5 OLE 对象流数据被写入临时文件“%TEMP%\Target.scr”

微软为了修补“沙虫”漏洞 CVE-2014-4114，发布了补丁 MS14-060[6]，在 CopyStreamToFile 之后增添了 MarkFileUnsafe 函数将该临时文件进行 MOTW 处理，将其 Security Zone 标记为不可信来源，尝试安装运行时会弹出 UAC 安全警告窗口[7]。

```
if ( result >= 0 )
{
  if ( CopyStreamToFile(
         NumberOfBytesWritten,
         *(LPCWSTR *)(*((_DWORD *)v2 + 16) + 592),
         *(_DWORD *)(*((_DWORD *)v2 + 16) + 68)) >= 0 )
  {
    MarkFileUnsafe(*(const unsigned __int16 **)(*((_DWORD *)v2 + 16) + 592));
    return 0;
  }
```

图 2-6 方法 CPackage::EmbedReadFromStream 增添了 MarkFileUnsafe 标记

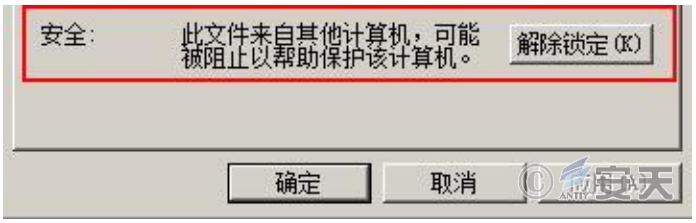


图 2-7 “%TEMP%\Target.scr”被标记为不可信

此时“%TEMP%\Target.scr”已经被释放在临时目录，如果我们以双击的方式打开该 PPSX 样本，自动播放模式将会开启，模块 packager.dll 的 CPackage::DoVerb 方法将检查样本指定 Verb 动作“cmd”的值，将其赋给变量 nVerb 和 nTrueVerb，来决定如何处理嵌入文件“Target.scr”。

这里常见的利用手法是构造“cmd”值为 3，这样就能模拟 popup 菜单操作执行右键菜单的第二项“以管理员权限运行” [8]，最终调用 SHELL32!CDefFolderMenu::InvokeCommand 方法执行“Target.scr”。

然而这里样本“\ppt\slides\ slide1.xml”指定 Verb 动作“cmd=0”，独辟蹊径采用了另一条流程。

```
</p:stCondLst><p:childTnLst><p:cmd type="verb" cmd="0"><p:cBhvr>
```

```
if ( nVerb == 2 )
  nTrueVerb = *((_DWORD *)this + 35);
if ( nTrueVerb == -1 || nTrueVerb == -2 )
  goto LABEL_8;
if ( nTrueVerb == 1 )
  return CPackage::_ChangePackageLabel((CPackage *)((char *)this - 8), a6);
if ( !nTrueVerb )
  return CPackage::_ExecuteAttachment(
           (CPackage *)((char *)this - 8),
           0,
           a3,
           (struct IOleClientSite *)hmenu,
           a5,
           a6,
           (const struct tagRECT *)v20);
```

调试器中我们也看到此时 nTureVerb 已被赋值为 0，开始执行函数 CPackage::_ExecuteAttachment。

```
7398587a 33ff          xor     edi,edi
7398587c 3bdf          cmp     ebx,edi
7398587e 751f          jne     packager!CPackage::DoVerb+0x126 (7398589f)
73985880 ffb5d0f9ffff  push    dword ptr [ebp-630h]
73985886 51            push    ecx
73985887 ff7518        push    dword ptr [ebp+18h]
7398588a 8d4ef8        lea     ecx,[esi-8]
7398588d ffb5d4f9ffff  push    dword ptr [ebp-62Ch] ss:002b:0032c42c=031065a0
73985893 52            push    edx
73985894 57            push    edi
73985895 e89c170000    call    packager!CPackage::_ExecuteAttachment (73987036)
```

判断nTrueVerb是否为零

进入 CPackage::_ExecuteAttachment 我们看到它调用了危险的 CPackage::_ActivateEmbeddedFile 方法，内部继续调用 shdocvw!CAttachmentServices::Execute 方法。

```
73987192 ffb570f7ffff  push    dword ptr [ebp-890h]
73987198 ffb558f7ffff  push    dword ptr [ebp-8A8h]
7398719e ffb56cf7ffff  push    dword ptr [ebp-894h]
739871a4 ff7514        push    dword ptr [ebp+14h]
739871a7 ffb55cf7ffff  push    dword ptr [ebp-8A4h]
739871ad ffb568f7ffff  push    dword ptr [ebp-898h]
739871b3 ff7508        push    dword ptr [ebp+8]
739871b6 57            push    edi
739871b7 e8e4deffff    call    packager!CPackage::_ActivateEmbeddedFile (739850a0)
```

```
0:000> t
eax=00596e40 ebx=0383a618 ecx=7395386c edx=001cba1c esi=00000000 edi=031c66e0
eip=73954f8c esp=001cb9dc ebp=001cba34 iopl=0         nv up ei pl zr na pe nc
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00000246
shdocvw!CAttachmentServices::Execute:
73954f8c 8bff            mov     edi,edi
0:000> kv
ChildEBP RetAddr  Args to Child
001cb9d8 73985311 00596e40 003c01c0 00000000 shdocvw!CAttachmentServices::Execute (FPO: [Non-Fpo])
001cba34 739871bc 0383a618 00000000 00000000 packager!CPackage::_ActivateEmbeddedFile+0x271 (FPO: [Non-Fpo])
001cc34c 7398589a 00000000 00000000 031c66e0 packager!CPackage::_ExecuteAttachment+0x186 (FPO: [Non-Fpo])
001cca00 70952fae 0383a620 00000000 00000000 packager!CPackage::DoVerb+0x121 (FPO: [Non-Fpo])
WARNING: Stack unwind information not available. Following frames may be wrong.
```

图 2-8 “cmd=0”时调用了危险的 CPackage::_ActivateEmbeddedFile 方法

shdocvw!CAttachmentServices::Execute 调用_OpZoneCheck 检查“%TEMP%\Target.scr”的 Security Zone 标记，最后调用_OpUserTrust 弹出 UAC 安全警告窗口提醒用户选择是否执行“%TEMP%\Target.scr”：

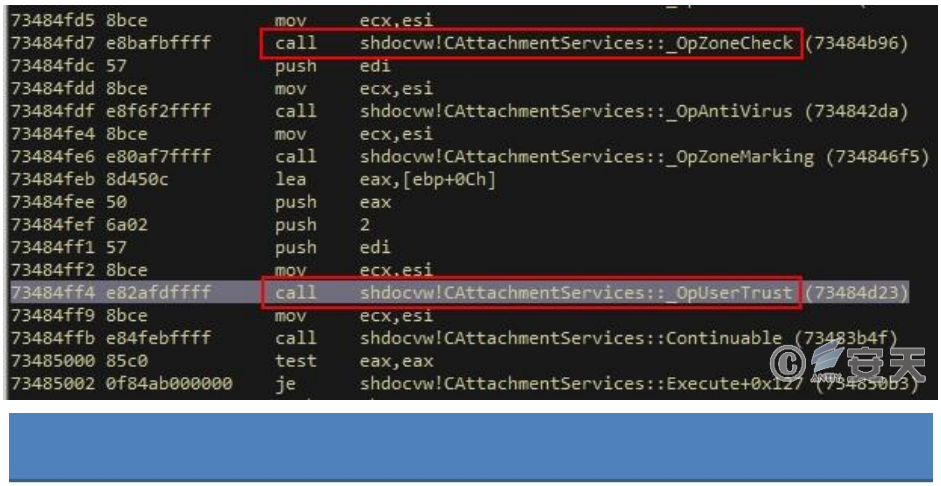


图 2-9 启动 Security Zone 标记检查，PPT 页面弹出 UAC 安全警告窗口

由于沙虫系列漏洞是一个链接执行漏洞，而非常见的格式文档溢出漏洞，DEP 等内存安全防御机制对其无效，调高 UAC 的级别对类似漏洞有一定的防御效果。如果此时被攻击者选择了继续运行“Target.scr”，恶意代码将被执行。而如果被攻击者的系统关闭了 UAC 控制或在获取了管理员权限的情况下，该安全警告窗口将不会弹出，“Target.scr”会被静默无警告地执行。

注：关于“沙虫”漏洞和其绕过漏洞 CVE-2014-6352 的更多细节可以参考安天[8]和 360 天眼实验室[9]及 McAfee[10]的分析文章。

2.2 Target.scr 样本分析

Target.scr 样本伪装成屏幕保护程序，由 Visual Basic 6.0 编写且未加壳，基本信息见表 2-2：

表 2-2 Target.scr 样本标签

病毒名称	Trojan/Win32. Barys
原始文件名	Target.scr
MD5	FFC6908F2D2990F9E0EF51194A387C4B
处理器架构	Intel 386 or later, and compatibles
文件大小	588.0 KB (602112 bytes)
文件格式	BinExecute/Microsoft.EXE[:X86]
时间戳	2018-10-03 13:54:37
加壳类型	未加壳
编译语言	Microsoft Visual Basic(6.0)[Native]
VT 首次上传时间	2018-10-15 06:10:27 UTC
VT 检测结果	29 / 67

通过反编译分析发现该样本的原始文件描述为"Gate Pass Management System　(Klash (Pvt) Ltd.)"：

```
Caption = "Gate Pass Management System  (Klash (Pvt) Ltd.)"
BackColor = &H8000000C&
WindowState = 2
'Icon = n/a
```

图 2-10 样本反编译后显示描述“Gate Pass Management System”

样本反编译代码的更多部分是并不能被调用的各种数据库操作功能函数，以及大量的数据库操作语句。这些数据库功能代码和语句能够连接数据库并对管理账户和库存记录进行增删改查。

例如函数 AddNewIGPass()和 DeleteIGPass()能通过 SQL 语句操作“tblIGMaster”和“tblIGDetails”表，新增和删除库存记录：

```
loc_0042765C: var_30 = txtIGNo.Text
loc_0042769D: var_38 = txtType.Text
loc_004276D5: var_34 = "INSERT INTO tblIGMaster VALUES(" & var_30

loc_00427C86: var_30 = txtIGNo.Text
loc_00427CB8: var_1A8 = var_18
loc_00427CD4: var_1E8 = var_18
loc_00427CEB: var_34 = "INSERT INTO tblIGDetails VALUES (" & var_30

loc_00428797: var_18 = txtIGNo.Text
loc_004287D2: var_20 = txtType.Text
loc_0042883D: var_28 = "DELETE FROM tblIGMaster WHERE igNo = " & var_18 & " and Type = '" & var_20
loc_00428854: var_2C = var_28 & var_0040DA24
loc_0042885A: var_E4 = var_28 & var_0040DA24
```

图 2-11 通过“INSERT”和“DELETE”语句增删库存记录

表 2-3 部分数据库被操作表名和操作语句一览

表名	操作语句
tblIGDetails	DELETE FROM tblIGDetails WHERE igNo =
tblIGDetails	INSERT INTO tblIGDetails VALUES (
tblMainAccounts	INSERT INTO tblMainAccounts VALUES(
tblMainAccounts	SELECT Acc_Code FROM tblMainAccounts WHERE Acc_Code =
tblPartyDetails	INSERT INTO tblPartyDetails VALUES(
tblPartyDetails	SELECT * FROM tblPartyDetails ORDER BY pCode DESC
tblIGDetails	SELECT igNo,Acc_Code,Sub_Acc, Nar, qtyin, qtyout, Unit FROM tblIGDetails WHERE igNo =

通过安天同源分析系统的比对，我们找出了对应的开源代码[2]，根据项目描述，这份代码适用于公司的库存管理系统：

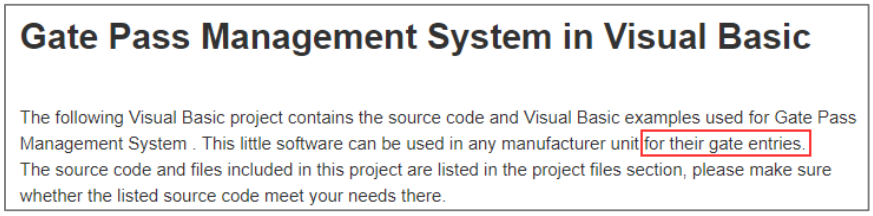


图 2-12 该 Visual Basic 项目的描述（这份代码适用于公司的库存管理系统）

该项目开发者是“Muhammad Assir Nadeem”，是巴基斯坦境内服装业公司 Klash Private Limited 的 IT 经理，根据我们的综合分析，没有理由怀疑该公司和项目开发者与攻击事件有关：

图 2-13 “IGOG Gate Pass System”开发者的个人介绍

攻击者并未利用开源代码中的任何功能调用，而是将加密后的 DarkComet 远控木马作为一个数据块嵌入到相关代码中，通过修改 Main 函数 Main_441770()的代码将木马运行起来，而开源代码的相关功能都不会被执行。关于使用正常数据库管理客户端源码进行修改的目的，根据 2.3 小节更多关联样本的分析，可以认为这是一种具有迷惑性的免杀方式。

函数 Main_441770()的修改主要是为了进行两个动作：

其一是实现将%TEMP%目录下的文件“Target.scr”复制一份至启动目录下，在主机侧建立持久化：

文件路径：“%AppData%\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\patakia.exe”

```
loc_0044192B: mov var_4, 00000007h
loc_00441932: mov var_E8, 004110C0h ; "Startup"
loc_00441977: push 0040F090h ; "SpecialFolders"
loc_00442326: var_6C = Trim(eax+0000000Ch)
loc_0044233F: var_E8 = "\patakia.exe"
loc_0044237A: var_2C = var_8C & "\patakia.exe"
loc_00442751: push eax
loc_00442752: call [00401194h] ; FileCopy %x1, %St,Var2
```

图 2-14 复制自身至启动目录下“patakia.exe”

“patakia.exe”的文件相关 Version 信息等，同公开的源码完全符合：

图 2-15 文件“patakia.exe”的描述

其二是实现解密资源节中存放的恶意代码并运行。样本在原有库管理客户端源码的基础上修改了执行逻辑，实际运行过程中解密运行资源节嵌入了加密的 Darkcomet 远控木马。“DarkComet RAT”又称“暗黑彗星”木马，具有远程操控、击键记录、摄像头监控、文件窃取和 DDoS 攻击等诸多丰富功能，是黑产活动中非常常见的木马类别。由于原有开源代码中的功能代码不会被调用执行，加之其文件拷贝的位置并不在对应的程序目录下，这基本排除掉了攻击者是为了替换原有主机环境的 IT 工具来建立持久化入口的猜测

样本“patakia.exe”的资源节（ID=109 处）包含了 32,219 字节的垃圾数据和 276,995 字节的加密数据（以[++...--...+]为标志头）：

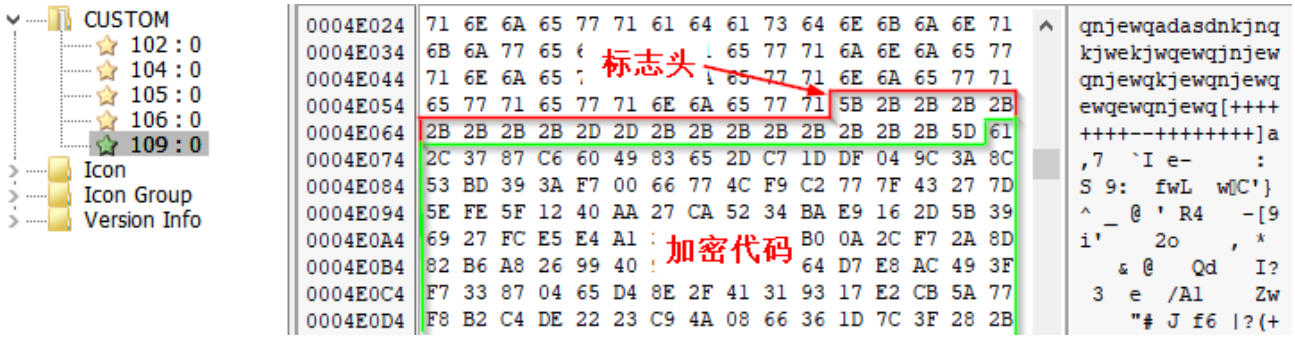


图 2-16 资源节 ID=109 处包含的加密数据

加密数据会在内存中被加载、解密，并展开成一个完整的 Darkcomet 木马 PE 文件：

```
var_F8 = "CUSTOM"
var_eax = Global.LoadResData(var_E4, var_0040E770, vbNullString)
var_114 = Global.LoadResData(var_E4, var_0040E770, vbNullString)
var_3C = "[++++++++--++++++++]"
var_E8 = var_3C
var_CC = Split(StrConv(var_CC, 64, 0), var_3C, -1, 0)
```


图 2-17 从资源节 ID=109 加载由标志头开始的数据

Address	Length	Type	String
.text:0047C5A0	0000000E	C	#BOT#VisitUrl
.text:0047C5B8	0000000D	C	#BOT#OpenUrl
.text:0047C634	0000000A	C	#BOT#Ping
.text:0047C67C	0000000F	C	#BOT#RunPrompt
.text:0047C6E0	00000011	C	#BOT#CloseServer
.text:0047C73C	00000012	C	#BOT#SvrUninstall
.text:0047C79C	0000000F	C	#BOT#URLUpdate
.text:0047C874	00000011	C	#BOT#URLDownload
.text:0047C890	00000031	C	#botCommand%Mass Download : Downloading File...

图 2-18 从内存中 dump 出的 PE 包含典型 Darkcomet 木马特征

木马尝试反向连接攻击者 C&C 服务器（desk1pc.ddns.net:700），其当前解析 IP185.82.***.***即为两封钓鱼邮件的发信 IP，可以认为攻击者掌控的资源相对有限。

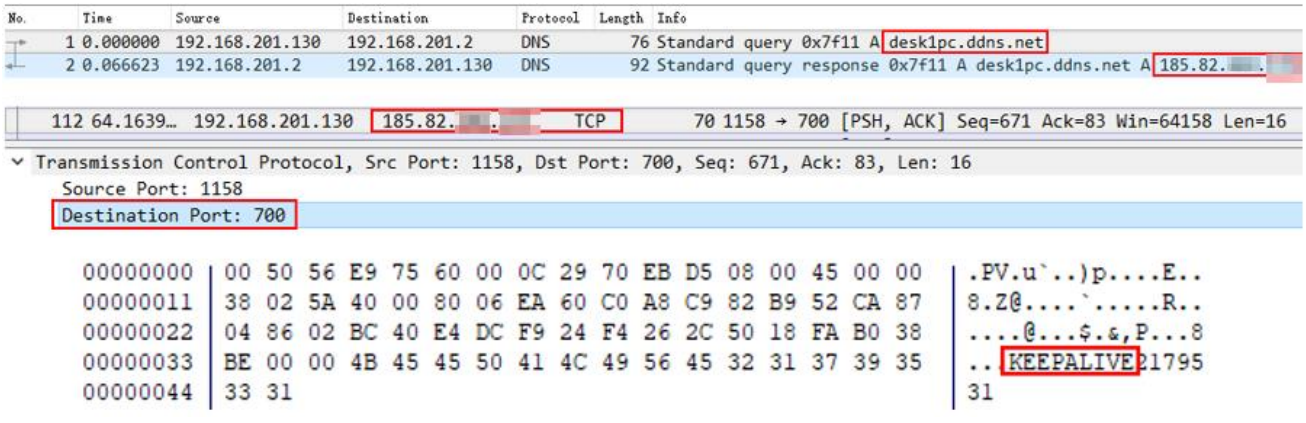


图 2-19 TCP 请求中包含的“KEEPALIVE+数字”用以维持同 C&C 的网络连接

2.3 扩线分析

根据本报告 1.1 节中的相关信息扩线可以找到其他一些手法如出一辙、攻击目标皆为银行的恶意邮件。但其并非从马拉维银行工作人员的信箱发出，而是盗用了其他金融机构的员工信箱，且邮件的发信方来自同一台攻击者主机。

其中一封信的收信人是马拉维国民银行的纳卡洛（nchalo）分行。时间最早的一封邮件通过被盗邮箱 Justin Akwuasaba <nigel.codrington@rbc.com>发送，该被盗邮箱系加拿大皇家银行（Royal Bank of Canada）的员工工作信箱。发信时间为“2018 年 6 月 22 日（周五）22:03”，发信 IP 为“185.82.***.***”，同 1.1 节中的发信 IP 185.82.***.***属同一 C 段。该恶意邮件还发送给了马拉维国民银行的另一待攻击目标 nchalo <nchalo@natbankmw.com>，其为马拉维国民银行纳卡洛（nchalo）分行的官方联系邮箱。

图 2-20 马拉维国民银行 nchalo 分行收到的攻击邮件

在攻击邮件中也有其他国家的银行目标，如其中一封邮件发送自邮箱“Audit Dept mgmt@prabhugroupusa.com”，系美国大型金融服务（银行间汇款转账）公司“Prabhu”[11]的员工工作邮箱。发信时间为“2018 年 8 月 25 日（周六） 10:19”，发信 IP 为“185.82.***.***”，与上述第一封邮件的发信 IP 一致。收信邮箱 yenisey.salahov@ibar.az 系阿塞拜疆国际银行（The International Bank of Azerbaijan）员工工作邮箱。

```
Received: from [redacted] (bjnmlumy.painreliv.uno [185.82.[redacted]])
    by mx.ibar.az with ESMTP id r7XoJFdU7oAlaMes for <yenisey.salahov@ibar.az>;
    Fri, 24 Aug 2018 23:18:38 +0400 (+04)
X-Barracuda-Envelope-From: mgmt@prabhugroupusa.com
X-Barracuda-Effective-Source-IP: bjnmlumy.painreliv.uno[185.82.[redacted]
X-Barracuda-Apparent-Source-IP: 185.82.[redacted]
From: "Audit Dept" <mgmt@prabhugroupusa.com>
Subject: DAILY REPORT ERROR
To: "yenisey.salahov" <yenisey.salahov@ibar.az>
```

图 2-21 阿塞拜疆国际银行收到的同类攻击邮件

这两封邮件的两个 PPSX 文件附件与 1.1 节中描述的样本一样，都利用了 CVE-2014-6352 漏洞，OLE 对象的“Ole10Native”流都包含了完整 PE 数据（“Document.scr”、“Document.exe”），原始保存路径则出现了新的路径“C:\Users\kboy\”。Verb 动作也都设置为“cmd=0”以执行临时目录下释放的绑定了恶意代码的正常程序。

```
loc_0076B016: var_58 = Trim(ecx+0000000Ch)
loc_0076B02F: var_CC = "\grfm.exe"
loc_0076B067: var_3C = var_80 & "\grfm.exe"

loc_005234D2: var_6C = Trim(var_90)
loc_005234E9: var_E8 = "\nazrein.exe"
loc_00523522: var_2C = var_8C & "\nazrein.exe"
```

图 2-22 释放的 PE 复制至启动目录后的文件名

样本“Document.scr”和“Document.exe”同 2.2 节中的“Target.scr”构造手法完全一致，它们分别在开源程序“Tech2ks NComputing Caffe[12]”和国际象棋游戏“ChessQuest”的源代码中插入 DarkComet 木马，甚至连木马的回连 C2 都共用着同一个域名和端口：“desk1pc.ddns.net:700”。以样本“Document.exe”为例，同 2.2 节中的样本“Target.scr”进行特征比对能展示出更多相似之处：

表 2-4 拓线样本特征对比

特征对比	样本“Target.scr”	样本“Document.scr”
源码	Gate Pass Management System in Visual Basic[2]	ChessQuest[获取来源未知]
编译语言	Microsoft Visual Basic(6.0)	Microsoft Visual Basic(6.0)
嵌入方式	PE 资源节 CUSTOM 目录下，ID=109 处，加密数据标志头：“[++++++++--++++++++]”	PE 资源节 CUSTOM 目录下，ID=109 处，加密数据标志头：“[####++++####]”
嵌入恶意代码类型	DarkComet 远控木马	DarkComet 远控木马
嵌入加密代码大小	276,995 字节	244,750 字节
恶意代码 C2	desk1pc.ddns.net:700	desk1pc.ddns.net:700

通过检查 185.82.***.***~***这一段 IP，我们发现 IP 185.82.***.***和 185.82.***.***于今年 10 月初曾被用于上文的三个 Visual Basic 样本和其它远控木马作回传 C&C 服务器（皆绑定过临时二级域名

kango.ddns.net、kinging.ddns.net、segun.ddns.net 和 desk1pc.ddns.net）。而 IP 185.82.***.***则于今年 10 月初被用于挂载知名勒索软件 GandCrab。

通过扩线分析可以看出，攻击者使用了多种公开源码程序作为掩护传播恶意代码，由于源程序代码并不能正常运行，而且所使用的开源程序并非常见金融场景下的应用或工具软件，所以我们认为攻击者并不是替换用户原有程序来实现持久化，而是以此作为免杀和干扰分析的技巧。

3 小结

从已有信息来看，本次马拉维国民银行及其他多国金融机构遭受的网络攻击均使用同一手法，这一系列网络攻击初步判断是以金融机构为目标，以金融资产侵害为目的的犯罪行为。目前除能判断马拉维银行相关信箱已被入侵并作为攻击跳板外，目前尚无法判断和知晓事件相关的几家分行客服员工是否已经触发攻击流程而造成了实际的损失。综合攻击者使用的漏洞、载荷、地址资源、社工技巧来看，系由一定技术能力的个人攻击者或犯罪团伙的可能性较大。

金融机构的运行高度依赖于信息系统，传统金融机构更多依靠物理空间安全手段保证信息系统安全。但在互联网时代，由于为大量互联网用户提供服务以及银行间跨国汇兑转账等业务，金融机构的信息系统已经是一个事实上存在着互联网侧多暴露面的系统。金融系统的服务网站、网银接口、人员使用的工作电子邮件信箱、跨行转账服务等，在已有攻击事件中，成为攻击的入口。尤其金融机构的电子邮件由于是直接暴露的信息资产，极易成为攻击者的攻击入口，同时因为其中的员工邮箱遭受攻击后，容易在内部实现基于邮件信任链的攻击，导致连锁反应的发生。

和安天既往复盘分析的针对金融机构的其他攻击——如针对在孟加拉国央行、越南先锋银行等多国 SWIFT 系统的网络攻击事件相比，本事件属于低水平攻击事件。攻击者并未使用 0day 漏洞，其攻击载荷也并无数字签名盗用等方式伪装，属于在基础结构安全和纵深防御层面可以有效对抗遏制的攻击。如果相关金融机构实施严格的安全策略，做好补丁升级和配置加固等基本工作，配套部署具有主动防御能力的终端防御软件并保持更新升级，就可以大大降低相关载荷的执行概率。本事件客观上代表了部分经济不发达国家的金融机构处于较低的防护水平之中，网络安全投入能力相对较差，本身抵御风险的能力也不足。网络攻击带来的风险，有可能转化为国家级的金融灾难。

我国当前整个金融信息化发展水平较快，在网络安全防护方面也积累了大量模式和经验，但同时也面临着更高的安全风险，需要应对超高能力国家/地区威胁行为体所发动的网络攻击。这些行为体具有坚定的攻击意志、能够承担巨大的攻击成本。在大规模工程体系的支撑下，进行体系化的攻击作业。TAO 组织攻击中东的最大 SWIFT 服务机构就是其中的代表事件。防御这一层面的网络攻击，需要以能力导向建设，按照“综合发展、深度结合、全面覆盖、动态综合”的安全理念，形成高水平的防御能力。在这个过程中，中国的能力型安全厂商，不仅要做好当下，也要逐步放眼全球，与中国金融机构共同积累沉淀安全防护理念、经验和解决方案，在未来对经济不发达国家的金融机构实现网络安全能力的支持和援助。

附录一：参考资料

[1] CVE: CVE-2014-6352

https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6352

[2] Freesourcecode：Gate Pass Management System in Visual Basic

http://freesourcecode.net/vbprojects/21006/Gate-Pass-Management-System--in-Visual-Basic#.W8PphtlR2Ht

[3] Natbankmw：马拉维银行官网

https://www.natbankmw.com

[4] 维基百科：National_Bank_of_Malawi

https://en.wikipedia.org/wiki/National_Bank_of_Malawi

[5] Malawi24：workers-steal-millions-from-national-bank

https://malawi24.com/2016/02/09/workers-steal-millions-from-national-bank/

[6] 微软：Microsoft Security Bulletin MS14-060 – Important

https://docs.microsoft.com/en-us/security-updates/securitybulletins/2014/ms14-060

[7] 微软：Microsoft TechNet

https://technet.microsoft.com/en-us/ms537628(v=VS.71)

[8] 安天：沙虫（CVE-2014-4114）相关威胁综合分析报告

http://www.antiy.com/response/cve-2014-4114.html

[9] Freebuf：CVE-2014-6352 漏洞及定向攻击样本分析

http://www.freebuf.com/vuls/106853.html

[10] McAfee：Bypassing Microsoft’s Patch for the Sandworm Zero Day: Even ‘Editing’ Can Cause Harm

https://securingtomorrow.mcafee.com/mcafee-labs/bypassing-microsofts-patch-for-the-sandworm-zero-day-even-editing-can-cause-harm/

[11] Prabhu:about us

https://www.prabhuonline.com/information/why-us

[12] Freesourcecode：Tech2ks NComputing Caffe in Visual Basic 较新版本

http://freesourcecode.net/vbprojects/17691/Tech2ks-NComputing-Caffe-in-Visual-Basic#.W_vKmLEzZaQ