Using PowerShell for Security Testing (2)

When performing a security test on/against a Microsoft Windows system we can make use of a series of PowerShell commands to profile and exploit the target system. The Get-LocalUser PowerShell cmdlet lists all the local users on a device.

```
PS C:\> get-localuser
Name               Enabled Description
----               ------- -----------
Administrator      False   Built-in account for administering the computer/domain
andre              True
Andrew Blyth       True
DefaultAccount     False   A user account managed by the system.
Guest              False   Built-in account for guest access to the
computer/domain
WDAGUtilityAccount False   A user account managed and used by the system for
Windows
PS C:\Users\Andrew Blyth> get-localuser -name "Andrew Blyth"
Name         Enabled Description
----         ------- -----------
Andrew Blyth True
PS C:\> get-localuser -name "Andrew Blyth" | select *
AccountExpires         :
Description            :
Enabled                : True
FullName               :
PasswordChangeableDate : 02/07/2022 21:09:44
PasswordExpires        :
UserMayChangePassword  : True
PasswordRequired       : False
PasswordLastSet        : 02/07/2022 21:09:44
LastLogon              : 19/08/2022 17:14:13
Name                   : Andrew Blyth
SID                    : S-1-5-21-1742287042-31044576-2166867149-1005
PrincipalSource        : Local
ObjectClass            : User
PS C:\>
```

We can also query a PowerShell to get information about a user if we know the SID associated with this user. This is illustrated in the following example.

```
PS C:\> get-localuser -SID S-1-5-21-1742287042-31044576-2166867149-500
Name          Enabled Description
----          ------- -----------
Administrator False   Built-in account for administering the computer/domain
PS C:\>
```

We can use PowerShell to access function on other computers. In the following example we are using PowerShell to list all of the user accounts on the computer **Desktop-01**. This is achieved via invoking the **Win32_UserAccount** WMI call object.

```
PS C:\ get-wmiobject -ComputerName Desktop-01 -Class Win32_UserAccount -Filter
"LocalAccount=True"
AccountType : 512
Caption     : DESKTOP-01\Administrator
Domain      : DESKTOP-01
SID         : S-1-5-21-1742287042-31044576-2166867149-500
FullName    :
Name        : Administrator
```