Dumping Active Directory Domain Info – with PowerUpSQL!

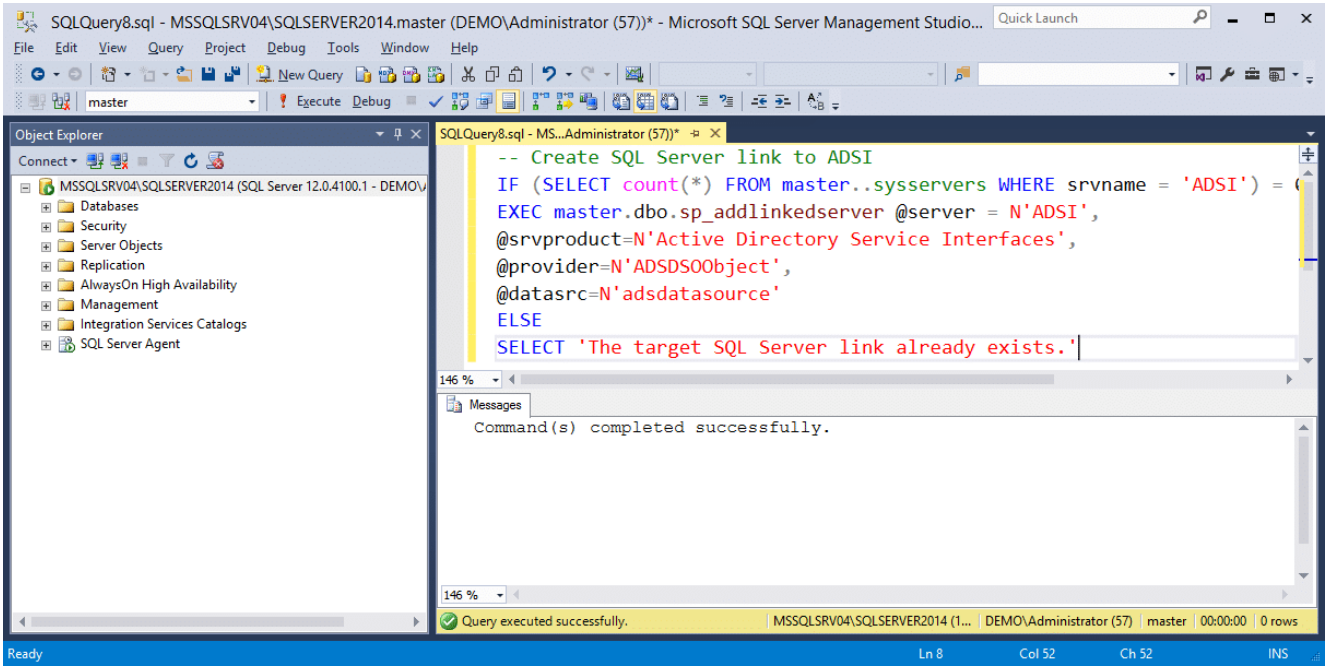
written by Thomas Elling | May 31, 2018

This blog walks through how to use the OLE DB ADSI provider in SQL Server to query Active Directory for information. I'll also share a number of new PowerUpSQL functions that can be used for automating common AD recon activities through SQL Server. Hopefully this will be useful to red teamers, pentesters, and database enthusiasts. Thanks to Scott Sutherland (@_nullbind) for his work on both the AD recon functions and PowerUpSQL!

The T-SQL

The T-SQL below shows how the ADSI provider is used with OPENQUERY and OPENROWSET to query for Active Directory information. First, a SQL Server link needs to be created for the ADSI provider. A link is created with the name "ADSI".

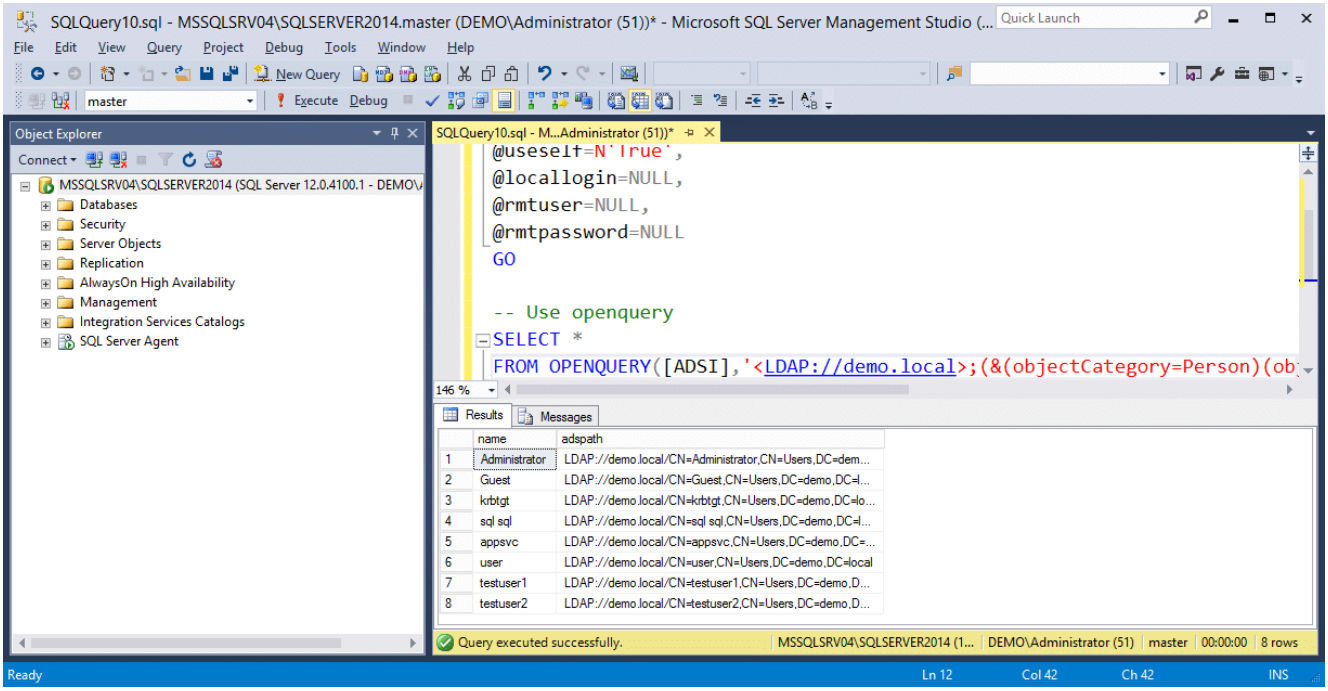
```
-- Create SQL Server link to ADSI
IF (SELECT count(*) FROM master..sysservers WHERE srvname =
'ADSI') = 0
EXEC master.dbo.sp_addlinkedserver @server = N'ADSI',
@srvproduct=N'Active Directory Service Interfaces',
@provider=N'ADSDSOObject',
@datasrc=N'adsdatasource'
ELSE
SELECT 'The target SQL Server link already exists.'
```



If using OPENQUERY, associate the link with the current authentication context. A username and password can also be specified here. Then run the example query.

Note: The LDAP “path” should be set to the target domain.

```
-- Define authentication context - OpenQuery
EXEC sp_addlinkedsrvlogin
@rmtsrvname=N'ADSI',
@useself=N'True',
@locallogin=NULL,
@rmtuser=NULL,
@rmtpassword=NULL
GO
-- Use openquery
SELECT *
FROM
OPENQUERY([ADSI],'<LDAP://path>;(&(objectCategory=Person)(objectClass=user));name, adspath;subtree')
```



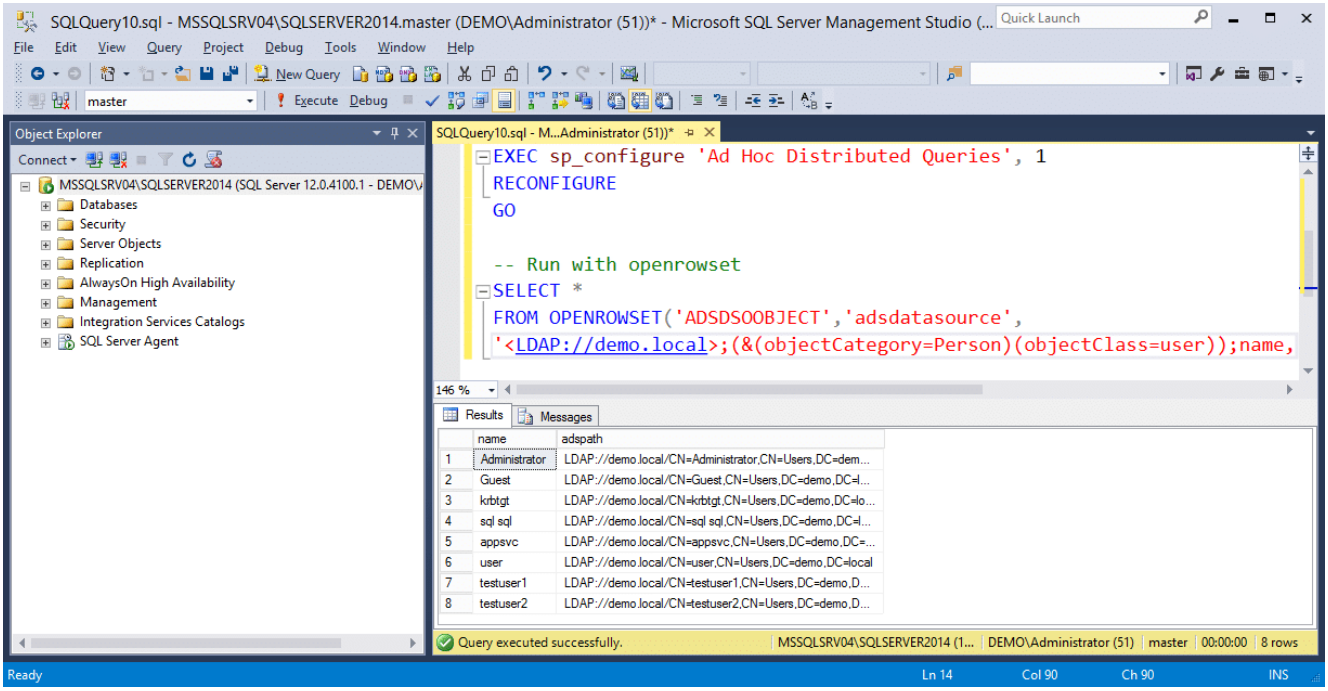
If using OPENROWSET, enable ad hoc queries. Then run the example query with a specified username and password or default authentication.

Note: The LDAP “path” should be set to the target domain.

```
-- Enable 'Show Advanced Options'
EXEC sp_configure 'Show Advanced Options', 1
RECONFIGURE
GO

-- Enable 'Ad Hoc Distributed Queries'
EXEC sp_configure 'Ad Hoc Distributed Queries', 1
RECONFIGURE
GO

-- Run with openrowset
SELECT *
FROM OPENROWSET('ADSDSOOBJECT','adsdatasource',
'<LDAP://path>;(&(objectCategory=Person)(objectClass=user));na
me, adspath;subtree')
```



Loading PowerUpSQL

PowerUpSQL can be loaded quite a few different ways in PowerShell. Below is a basic example showing how to download and import the module from GitHub.

```
IEX(New-Object System.Net.WebClient).DownloadString("https://raw.githubusercontent.com/NetSPI/PowerUpSQL/master/PowerUpSQL.ps1")
```

Newly Added Active Directory Recon Functions

Now that you have PowerUpSQL loaded, you can use the new commands to execute queries against the domain. However, please be aware that all commands require sysadmin privileges.

Function Name	Description
Get-SQLDomainAccountPolicy	Provides the domain account policy for the SQL Server's domain.

Function Name	Description
Get-SQLDomainComputer	Provides a list of the domain computers on the SQL Server’s domain.
Get-SQLDomainController	Provides a list of the domain controllers on the SQL Server’s domain.
Get-SQLDomainExploitableSystem	Provides a list of the potential exploitable computers on the SQL Server’s domain based on Operating System version information.
Get-SQLDomainGroup	Provides a list of the domain groups on the SQL Server’s domain.
Get-SQLDomainGroupMember	Provides a list of the domain group members on the SQL Server’s domain.
Get-SQLDomainObject	Can be used to execute arbitrary LDAP queries on the SQL Server’s domain.
Get-SQLDomainOu	Provides a list of the organization units on the SQL Server’s domain.
Get-SQLDomainPasswordsLAPS	Provides a list of the local administrator password on the SQL Server’s domain. This typically required Domain Admin privileges.
Get-SQLDomainSite	Provides a list of sites.
Get-SQLDomainSubnet	Provides a list of subnets.

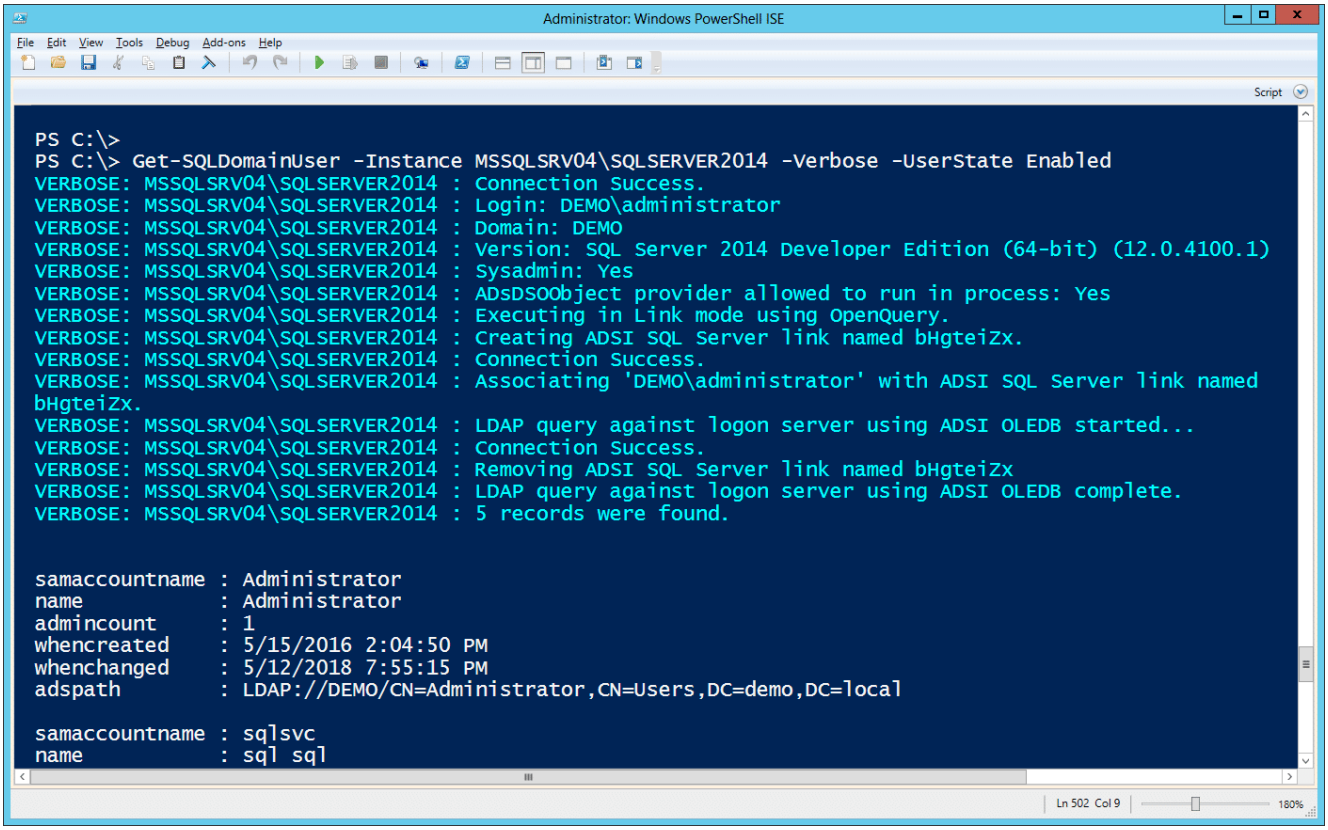
Function Name	Description
Get-SQLDomainTrust	Provides a list of domain trusts.
Get-SQLDomainUser	Provides a list of the domain users on the SQL Server’s domain.
Get-SQLDomainUser -UserState Disabled	Provides a list of the disabled domain users on the SQL Server’s domain.
Get-SQLDomainUser -UserState Enabled	Provides a list of the enabled domain users on the SQL Server’s domain.
Get-SQLDomainUser -UserState Locked	Provides a list of the locked domain users on the SQL Server’s domain.
Get-SQLDomainUser -UserState PreAuthNotRequired	Provides a list of the domain users that do not require Kerberos preauthentication on the SQL Server’s domain.
Get-SQLDomainUser -UserState PwLastSet 90	This parameter can be used to list users that have not changed their password in the last 90 days. Any number can be provided though.
Get-SQLDomainUser -UserState PwNeverExpires	Provides a list of the domain users that never expire on the SQL Server’s domain.
Get-SQLDomainUser -UserState PwNotRequired	Provides a list of the domain users with the PASSWD_NOTREQD flag set on the SQL Server’s domain.

Function Name	Description
Get-SQLDomainUser -UserState PwStoredRevEnc	Provides a list of the domain users storing their password using reversible encryption on the SQL Server’s domain.
Get-SQLDomainUser -UserState SmartCardRequired	Provides a list of the domain users that require smart card for interactive login on the SQL Server’s domain.
Get-SQLDomainUser -UserState TrustedForDelegation	Provides a list of the domain users trusted for delegation on the SQL Server’s domain.
Get-SQLDomainUser -UserState TrustedToAuthForDelegation	Provides a list of the domain users trusted to authenticate for delegation on the SQL Server’s domain.

Dumping Domain Users Examples

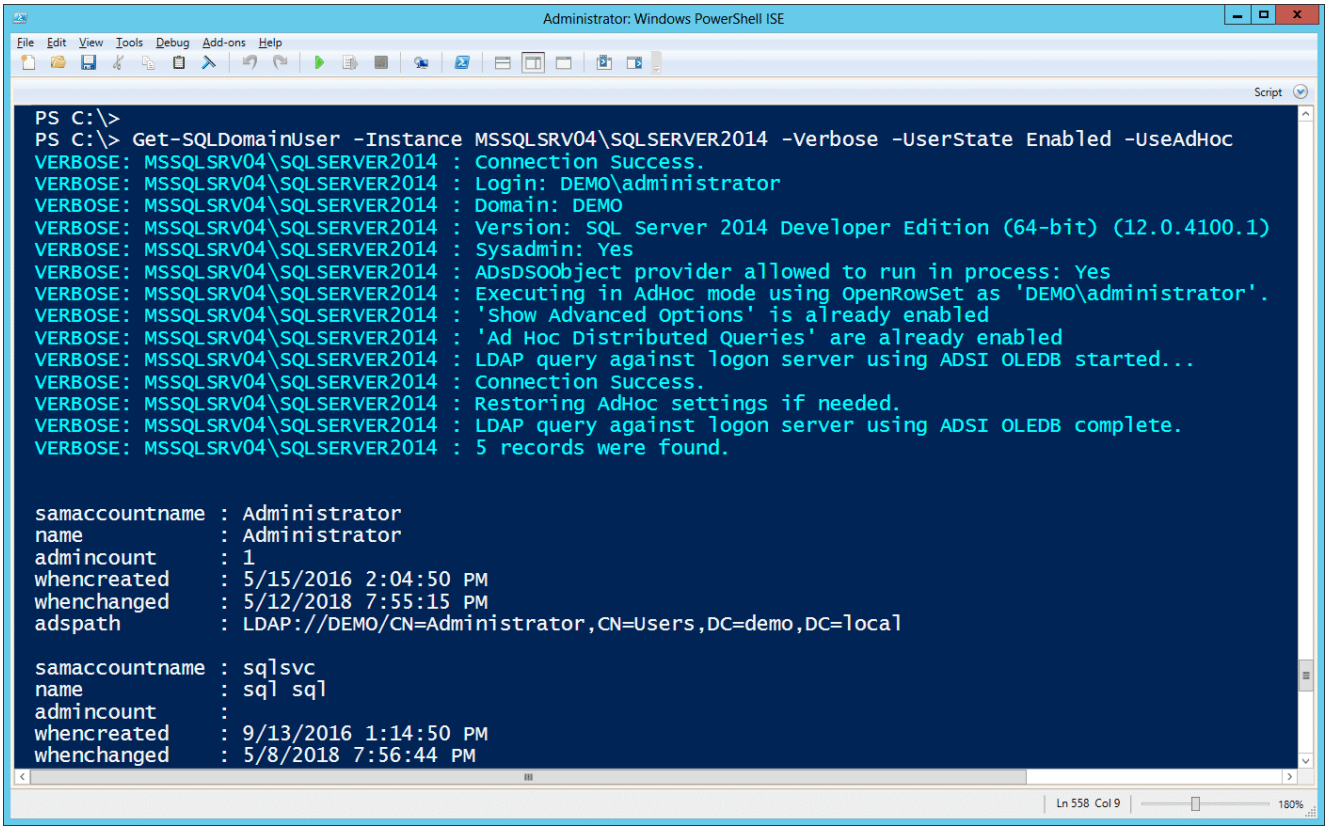
This example shows how to gather a list of enabled domain users using a Linked Server via OPENQUERY.

```
Get-SQLDomainUser -Instance MSSQLSRV04SQLSERVER2014 -Verbose -UserState Enabled
```



Alternatively, the command can be run using ad hoc queries via OPENROWSET as shown below. Its nothing crazy, but it does provide a few options for avoiding detection if the DBAs are auditing for linked server creation, but not ad hoc queries in the target environment.

```
Get-SQLDomainUser -Instance MSSQLSRV04SQLSERVER2014 -Verbose -UserState Enabled -UseAdHoc
```



The functions also support providing an alternative SQL Server login for authenticating to the SQL Server and an alternative Windows credential for configuring server links. More command examples can be found here.

The Authentication and Authorization Matrix

Depending on the current user's security context or the provided credentials, the user may not have access to query AD for information. The tables below illustrate privileges and the corresponding access.

OPENQUERY (Linked server) auth table by Scott Sutherland (@_nullbind)

Current User (Domain User – Public)	Current User (Domain User – Sysadmin)	Current User (SQL Login – Public)	Current User (SQL Login – Sysadmin)	Provided Domain User	Access
X					No
	X				Yes
		X			No
			X		No
X				X	No
	X			X	Yes
		X		X	No
			X	X	Yes

OPENROWSET (Ad Hoc query) auth table by Scott Sutherland (@_nullbind)

Current User (Domain User – Public)	Current User (Domain User – Sysadmin)	Current User (SQL Login – Public)	Current User (SQL Login – Sysadmin)	Provided Domain User	Access
X					No
	X				Yes
		X			No
			X		Yes
X				X	No
	X			X	Yes
		X		X	No
			X	X	Yes

Conclusion

Recon is an essential first step in assessing the security of an Active Directory environment. Thanks to some great work by Will Schroeder (@harmj0y) and others on Powerview. Hopefully these AD recon functions will provide another medium to accomplish the same end. For more information on the newly added AD recon functions, check out the PowerUpSQL wiki!