SOP Reporting on printer security

In the Windows operating system, all objects secured by the OS have four key properties:

The owner

The primary group

Discretionary Access Control List (DACL)   
System Access Control List (SACL)

The DACL contains a set of individual permissions, known as Access Control Entries   
(ACEs), that define a particular permission. Each ACE contains properties that describe the permission, including a trustee (the security principal to whom you are giving this   
permission), a permission mask (what permission is being allowed or disallowed), and an ACE type (what type is allowed, disallowed). You can find details of the permission masks on the MSDN.

Getting ready

This recipe displays the DACL for the Sales Group printer, SGCP1, created by the *Installing* *and sharing printers* recipe and later updated by the *Changing printer drivers* recipe. You could easily convert this recipe into an advanced function (for example, Get-PrinterSecurity) with a parameter to tell the function which printer to examine.

How to do it:

1.Create a hash table containing printer permissions:

$Permissions = @{

ReadPermissions = [uint32] 131072   
 Print = [uint32] 131080   
 PrintAndRead = [uint32] 196680   
 ManagePrinter = [uint32] 983052   
 ManageDocuments = [uint32] 983088   
 ManageChild = [uint32] 268435456   
 GenericExecute = [uint32] 536870912   
 ManageThisPrinter = [uint32] 983116

}

2.Get a list of all printers and select the Sales Group color printer:

$Printer = Get-CimInstance -Class Win32\_Printer `   
 -Filter "Name = 'SGCP1'"

3.Get the SecurityDescriptor and DACL for each printer:

$SD = Invoke-CimMethod -InputObject $Printer `   
 -MethodName   
 GetSecurityDescriptor   
 $DACL = $SD.Descriptor.DACL

4.For each Ace in the DACL, look to see what permissions you have set, and report   
accordingly:

ForEach ($Ace in $DACL) {

5.Look at each permission that can be set and check to see if the Ace is set for that permission:

Foreach ($Flag in ($Permissions.GetEnumerator() ) ) {   
 # Is this flag set in the access mask?

If ($Flag.value -eq $Ace.AccessMask) {

6.If this permission is set, then get the AceType:

$AceType = switch ($Ace.AceType)   
 {

0 {'Allowed'; Break}

1 {'Denied'; Break}

2 {'Audit'}

}

7.Get the permission type, nicely formatted:

$PermType = $flag.name -Csplit '(?=[A-Z])' -ne '' -join ' '

8.Finally, display the results (and end the loops and If statement):

'Account: {0}{1} - {2}: {3}' -f $ace.Trustee.Domain,   
 $Ace.Trustee.Name,   
 $PermType, $AceType

} # End of If $flag,Value

} # End Foreach $Flag loop

} # End Each $Ace

How it works...

This recipe begins, in *step 1*, by defining a hash table of the permissions that you can use in a printer's DACL. In *step 2*, you use the Get-CimInstance cmdlet to retrieve the WMI object   
relating to the Sales Group color printer.

In step 3, you use the GetSecurityDescriptor method of the printer object to get the   
DACL for this printer. The DACL, which you store in the $DACL variable, is an array of   
individual Win32\_ACE objects.

In *steps 4* you examine each Ace in the DACL to get, decode, and display the details of the   
permission expressed by this Ace entry. In *step 5*, you iterate through the permissions (as   
defined in *step 1*). In *step 6*, you check to see if the flag matches the AccessMask property of

the Ace. If the entry matches, you determine the ace type in *step 6*. In *step 7*, you get the   
permission type nicely formatted. Finally, in *step 8*, you display the particular permissions. The output from the final step in this recipe looks like this:

