```
'-------------------------------------------------------------------------------------------------------
'Use Case: #?  : Secedit Test
'Goal: Tests the attributes of one or more Security settings available within Secedit
'relation document : ????.doc
'Version: 1   Feb 3, 2006
'Author: McKee, Troy
'-------------------------------------------------------------------------------------------------------
'Set variables
'Option Explicit
Dim strComputer, objWMIService, colOSes, OSVer
Dim WshShell, objExecObject
Dim objFSO, objTextFile, strContents
Dim strFullInfName, strFullSdbName, strFullRptName, strFullRptNameLast
Dim strFileName, strPath, strConfig, strAnalyze
Dim strStatus,strTestResult, strTestResultMessage, strTestData, strResults, strErrMessage
Dim Attribute_value, operator, file_location, inf_name, sdb_name, negate_result, pass_notfound
Dim strLine

strStatus=""
strTestResult=""
strTestResultMessage=""
strTestData=""
strResults=""
strErrMessage=""

Const ForReading = 1
Const ForWriting = 2

strStatus = ""
strLine = ""

strComputer = "."

'On Error Resume Next

'-------------------------------------------------------------------------------------------------------
        'Attribute_value = objArguments(0)              'String representing the secedit attribute to test
        'operator = objArguments(1)                     'Options are Mismatch, Not Configured, or Not Available
        'file_location = objArguments(2)                'full path to .inf file to use (i.e. c:\windows\security\templates)
        'inf_name = objArguments(3)                     'name of file for security template (i.e. securews.inf)
        'sdb_name = objArguments(4)                     'name of file for security database (i.e. securews.sdb).
        'negate_result = objArguments(5)                'Negate Result
        'pass_notfound = objArguments(6)                'Behavior if the attribute not found. True = pass, false = fail

'-------------------------------------------------------------------------------------------------------
'Check OS version
Set objWMIService = GetObject("winmgmts:" _
 & "{impersonationLevel=impersonate}!\\" & strComputer & "\root\cimv2")

Set colOSes = objWMIService.ExecQuery("Select * from Win32_OperatingSystem")
```

```
For Each objOS in colOSes
  If objOS.BuildNumber = 3790 Then
        OSVer = "/import"
        'wscript.Echo "Server 2003"
  Else
        OSVer = "/config"
        'wscript.Echo "XP"
  End If
Next

Set objArguments = WScript.Arguments

If VerifyParameter(objArguments) Then

  'Create secedit command and parameters
  Set objFSO = CreateObject("Scripting.FileSystemObject")
  strFullsdbName = objFSO.BuildPath(file_location, sdb_name)
  strFullInfName = objFSO.BuildPath(file_location, inf_name)
  strFullRptName = ObjFSO.BuildPath(file_location, "secrpt.txt")
  strFullRptNameLast = ObjFSO.BuildPath(file_location, "secrpt-last.txt")
  strConfig = "secedit " & OSVer & " /db " & strFullSdbName & " /cfg " &
strFullInfName & " /log " & strFullRptName
  strAnalyze = "secedit /analyze /db " & strFullSdbName & " /cfg " & strFullInfName
& " /log " & strFullRptName

  bolTestPassed=""
  strErrMessage=""
  strTestResultMessage=""
  blnStatus=true

  If objFSO.FileExists(strFullRptName) Then
    objFSO.DeleteFile(strFullRptName)
  End If

  '-- create the shell which will run secedit.exe --
    If objFSO.FileExists(strFullsdbName) Then
        strStatus="success"
    Else
        If objFSO.FileExists(strFullInfName) Then  'Run Secedit to create .sdb file
          Set WshShell = WScript.CreateObject("WScript.Shell")
          Set objExecObject = WshShell.Exec(strConfig)
          objExecObject.StdOut.close
          strStatus="success"
        Else
          strStatus="failure"
          strTestResultMessage="Unable to obtain security template info"
          strErrMessage="Unable to obtain security template info"
        End If
        Wscript.Sleep 3000
    End If

    If objFSO.FileExists(strFullsdbName) Then 'Run Secedit to create log of results
        Set WshShell = WScript.CreateObject("WScript.Shell")
        Set objExecObject = WshShell.Exec(strAnalyze)
        objExecObject.StdOut.close
        strStatus="success"
    Else
        strStatus="failure"
        strTestResultMessage="Unable to obtain security database info"
        strErrMessage="Unable to obtain security database info"
    End If

  Wscript.Sleep 3000 'Gives time for file to be closed
```

```
    If objFSO.FileExists(strFullRptName) Then

        Set FS=CreateObject("Scripting.FileSystemObject")
        Set objTextFile = FS.GetFile(strFullRptName).OpenAsTextStream(1,-2)

        Do Until objTextFile.AtEndOfStream
          strLine = objTextFile.ReadLine

          If InStr(strLine, operator) = 1 Then   'Parse secedit log for comparison
to attribute_value
                If InStr(1, strLine, Attribute_value) And Not pass_notfound Then
                  strStatus="success"
                  strTestResult="fail"
                  strTestResultMessage="The system does not match the policy"
                  strTestData="The system does not match the policy"

                End If

          End If

          If strStatus="success" Then
            If strTestResult<>"fail" And Not negate_result Then
              strTestResult="pass"
              strTestResultMessage="The system matches the policy"
              strTestData="The system matches the policy"
            End If
          End If

        Loop

        objTextFile.Close

    Else
        strStatus="failure"
        strTestResultMessage="Unable to parse security report."
        strErrMessage="Unable to parse security report."
    End If

  Set objFSO = CreateObject("Scripting.FileSystemObject")
  If objFSO.FileExists(strFullRptNameLast) Then   'check if secedit-last.txt exists
    objFSO.DeleteFile(strFullRptNameLast)        'If so, delete
  End If

  objFSO.MoveFile strFullRptName , strFullRptNameLast    'rename secedit.txt to
secedit-last.txt

End If

strResults =
CreateReturnData(strStatus,strTestResult,strTestResultMessage,strTestData,strErrMess
age)
WScript.Echo (strResults)
'-------------------------------------------------------------------------------
-----------------------------------------------------

Function
CreateReturnData(strStatus,strTestResult,strTestResultMessage,strTestData,strErrMess
age)
        Dim strReturnXML
        strReturnXML =                "<ScriptOutput>"
        strReturnXML = strReturnXML & "<Status>" & strStatus & "</Status>"
        If strErrMessage <> "" Then
```

```
                strReturnXML = strReturnXML & "<ErrorMessages><message>"
&strErrMessage& "</message></ErrorMessages>"
        End If

        strReturnXML = strReturnXML & "<TestResult>" & strTestResult &
"</TestResult>"
        strReturnXML = strReturnXML & "<TestResultMessage>" & strTestResultMessage &
"</TestResultMessage>"
        strReturnXML = strReturnXML & "<TestData>" & strTestData  & "</TestData>"
        strReturnXML = strReturnXML & "</ScriptOutput>"

        CreateReturnData = strReturnXML
End Function

'-------------------------------------------------------------------------------
-------------------------------------------------------

Function VerifyParameter(objParameter)
        Dim iNumberOfArguments
        Dim blnRes
        blnRes = true
        iNumberOfArguments = objParameter.Count
        If      iNumberOfArguments<>7 Then
                blnRes=false
                strErrMessage="Parameter does not match. Usage: secedit.vbs
Attribute_value operator file_location inf_name sdb_name negate_result
pass_notfound"
        Else
                Attribute_value = objArguments(0)               'String representing
the secedit attribute to test
                operator = objArguments(1)                      'Options are
Mismatch, Not Configured, or Not Available
                file_location = objArguments(2)                 'full path to .inf
file to use (i.e. c:\windows\security\templates)
                inf_name = objArguments(3)                      'name of file for
security template (i.e. securews.inf)
                sdb_name = objArguments(4)                      'name of file for
security database (i.e. securews.sdb).
                negate_result = objArguments(5)                 'Negate Result
                pass_notfound = objArguments(6)                 'Behavior if the
services not found. True = pass, false = fail

        End If

       If blnRes Then
               Select Case operator
                       Case  "Mismatch"
                       Case  "Not Configured"
                       Case  "Not Available"
                       Case  Else
                              blnRes=false
                              strErrMessage="Unrecognized operator :'" & operator &
"' , For Parameter operator it only can be set to [Mismatch/Not Configured/Not
Available]"
              End  Select
        End If

        If blnRes Then
                Select Case negate_result
                       Case "true"
                       Case "false"
                       Case Else
                               blnRes=false
```

```
                                strErrMessage="Unrecognized negate_result value : "
& negate_result & " For Parameter negate_result, it only can be set to true/false"
                End Select
        End If

        If blnRes Then
                Select Case pass_notfound
                        Case "true"
                        Case "false"
                        Case Else
                                blnRes=false
                                strErrMessage="Unrecognized pass_notfound value : "
& pass_notfound & " For Parameter pass_notfound, it only can be set to true/false"
                End Select
        End If



        VerifyParameter=blnRes

End Function
```