

Листинг

Сценарий FirewallAuditReport.vbs

```

WScript.Echo VbCrLf & "Firewall Audit Report" & VbCrLf & String(21, "=") & VbCrLf
' Определяем для какого профиля пользователь хочет проводить аудит.
If WScript.Arguments.Count = 0 Then
    iType=2
Else
    Select Case LCase(WScript.Arguments(1))
        Case "domain"
            iType=0
        Case "standard"
            iType=1
        Case "current"
            iType=2
        Case Else
            WScript.Echo "Usage: fwAudit.vbs [standard|domain|current]"
            Stop
    End Select
End If

```

A ' Создаем объект локальной политики.
Set objPol = CreateObject("HNetCfg.FwMgr").LocalPolicy.GetProfileByType(iType)
' Определяем тип объекта политики, для указания в отчете
' поэтому если пользователь указывает параметр "Current," мы можем указать тип.
Select Case objPol.Type
 Case 0
 sType = "Domain Policy"
 Case 1
 sType = "Standard Policy"
 Case Else
 sType = "Other (" & objPol.Type & ")"
End Select
' Показываем глобальные настройки текущего профиля.
WScript.Echo "Global Settings:"
WScript.Echo GetCheck(objPol.FirewallEnabled) & "Firewall enabled"
WScript.Echo GetCheck(objPol.ExceptionsNotAllowed) & "Don't allow exceptions"
WScript.Echo GetCheck(Not objPol.NotificationsDisabled) & "Notifications" & VbCrLf

B ' Выводим список всех сетевых интерфейсов и информацию об состоянии сетевого экрана для каждого из них.
WScript.Echo "Per-interface settings:"
Set objNSH = Wscript.CreateObject("HNetCfg.HNetShare.1")
For Each objNetConn In objNSH.EnumEveryConnection
 Set objNShCfg = objNSH.INetSharingConfigurationForINetConnection(objNetConn)
 Set objNCProps = objNSH.NetConnectionProps(objNetConn)
 Wscript.echo GetCheck(objNShCfg.InternetFirewallEnabled) & " " & _
 objNCProps.Name & " — " & _
 objNCProps.DeviceName
Next

C ' Создаем заголовок таблицы для исключений сетевого экрана.
WScript.Echo VbCrLf & VbCrLf & "Exceptions" & VbCrLf & String(10, "=") & VbCrLf
WScript.Echo " X Name" & Space(30) & "Ports" & Space(6) & "Scope"
WScript.Echo String(79, "-")
' Выводим все исключения для служб.
For Each objService In objPol.Services
 sLine = GetCheck(objPol.FirewallEnabled) & " " & _
 FormatColumn(objService.Name, 34)
' Определяем какие порты используются службой.
 For Each oPort In objService.GloballyOpenPorts
 sLine = sLine & _
 FormatColumn(GetProtocol(oPort.Protocol) & " " & _
 oPort.Port, 11) & _
 FormatColumn(GetScope(oPort), 22) & VbCrLf & Space(38)
 Next
 sLine=Trim(sLine)
 WScript.Echo Left(sLine, Len(sLine)-1)
Next
' Выводим все исключения для приложений.
Set objFS = CreateObject("Scripting.FileSystemObject")
For Each objApp In objPol.AuthorizedApplications
 sLine = GetCheck(objApp.Enabled) & " "
 If objFS.FileExists(objApp.ProcessImageFileName) Then
 sLine = sLine & FormatColumn(objApp.Name, 34)
 Else
 sLine = sLine & FormatColumn(objApp.Name & " ** Missing **", 34)
 End If
 ' Приложения открывают все порты, нам необходимо указать только область применения.
 sLine = sLine & FormatColumn("Any", 11) & GetScope(objApp)
 WScript.Echo sLine
Next
' Выводим все исключения для портов.

```

For Each oPort in objPol.GloballyOpenPorts
    sLine = GetCheck(oPort.Enabled) & " " & _
        FormatColumn(oPort.Name, 34)
    sLine = sLine & FormatColumn(GetProtocol(oPort.Protocol) & " " & _
        oPort.Port, 11) & FormatColumn(GetScope(oPort), 22)
    WScript.Echo sLine
Next

```

D ' Выводим все схемы портов для интерфейсов.

```

WScript.Echo VbCrLf & VbCrLf & "Interface-Specific Port Mappings" & VbCrLf & _
String(32,"=") & VbCrLf
Set objNSH = Wscript.CreateObject("HNetCfg.HNetShare.1")
For Each objNetConn In objNSH.EnumEveryConnection
    Set objNSHCfg = objNSH.INetSharingConfigurationForNetConnection(objNetConn)
    Set objNCProps = objNSH.NetConnectionProps(objNetConn)
    Set colPortMaps = objNSHCfg.EnumPortMappings(0)
    Next
    ' Проверяем, включены ли схемы портов на данном интерфейсе
    ' если нет, то мы не отображаем интерфейс в отчете.
    iPortMapCount = 0
    For Each objPortMap in colPortMaps
        If objPortMap.Properties.Enabled = True Then iPortMapCount = _
iPortMapCount + 1
    Next
    ' Выводим все схемы портов для данного интерфейса.
    If iPortMapCount > 0 Then
        WScript.Echo VbCrLf & objNCProps.Name & " — " & _
objNCProps.DeviceName & VbCrLf
        WScript.Echo "Name" & Space(17) & "External Port" & Space(8) & _
"Internal Port" & Space(4) & "Internal IP Address"
        WScript.Echo String(79,"-")
        Set colPortMaps = objNSHCfg.EnumPortMappings(0)
        For Each objPortMap in colPortMaps
            If objPortMap.Properties.Enabled = True Then
                If objPortMap.Properties.IPProtocol=6 Then
                    sProto="TCP"
                Else
                    sProto="UDP"
                End If
                WScript.Echo FormatColumn(objPortMap.Properties.Name, 21) & _ FormatColumn(GetProtocol(objPortMap.
Properties.IPProtocol) & " " & "Port " & objPortMap.Properties.ExternalPort, 21) & _ FormatColumn(objPortMap.Properties.
InternalPort, 17) & _ FormatColumn(objPortMap.Properties.TargetIPAddress, 20)
            End If
        Next
    End If
Next

```

' Переводим номер протокола в формат строки.

```

Function GetProtocol(Protocol)
    Select Case Protocol
        Case 6
            GetProtocol = "TCP"
        Case 17
            GetProtocol = "UDP"
        Case Else
            GetProtocol = "Other"
    End Select
End Function
' Формируем строку с областью применения.
Function GetScope(oPort)
    Select Case oPort.Scope
        Case 0
            GetScope = "Any computer"
        Case 1
            GetScope = "Local subnet"
        Case 2
            GetScope = oPort.RemoteAddresses
    End Select
End Function
' Формируем текст с определенной шириной столбца.
Function FormatColumn(sText, iLength)
    If Len(sText) >= iLength Then
        FormatColumn = Left(sText, iLength — 1) & " "
    Else
        FormatColumn = sText & Space(iLength — Len(sText))
    End If
End Function
' Функция конвертации булевой переменной в поле выбора.
Function GetCheck(Condition)
    GetCheck = "[" & Chr(-(56 * Condition) + 32) & "]"
End Function

```