```
Imports System.IO
Imports System.Collections.Generic
Imports System.Text
Imports System.Runtime.InteropServices
Imports System.Security.AccessControl
Imports Microsoft.VisualBasic
Imports System.Web
Imports System.Security.Principal

Public Class ntfs_checker
    Implements IHttpModule
    <DllImport("advapi32.dll", SetLastError:=True)> _
    Private Shared Function GetEffectiveRightsFromAcl(pDacl As IntPtr, ByRef pTrustee As
    TRUSTEE, ByRef pAccessRights As ACCESS_MASK) As UInteger
    End Function

    <StructLayout(LayoutKind.Sequential, CharSet:=CharSet.Auto, Pack:=4)> _
    Private Structure TRUSTEE
        Private pMultipleTrustee As IntPtr
        ' must be null
        Public MultipleTrusteeOperation As Integer
        Public TrusteeForm As TRUSTEE_FORM
        Public TrusteeType As TRUSTEE_TYPE
        <MarshalAs(UnmanagedType.LPStr)> _
        Public ptstrName As String
    End Structure

    Private Enum TRUSTEE_FORM
        TRUSTEE_IS_SID
        TRUSTEE_IS_NAME
        TRUSTEE_BAD_FORM
        TRUSTEE_IS_OBJECTS_AND_SID
        TRUSTEE_IS_OBJECTS_AND_NAME
    End Enum

    Private Enum TRUSTEE_TYPE
        TRUSTEE_IS_UNKNOWN
        TRUSTEE_IS_USER
        TRUSTEE_IS_GROUP
        TRUSTEE_IS_DOMAIN
        TRUSTEE_IS_ALIAS
        TRUSTEE_IS_WELL_KNOWN_GROUP
        TRUSTEE_IS_DELETED
        TRUSTEE_IS_INVALID
        TRUSTEE_IS_COMPUTER
    End Enum

    <DllImport("advapi32.dll", CharSet:=CharSet.Auto)> _
    Private Shared Function GetNamedSecurityInfo(pObjectName As String, ObjectType As
    SE_OBJECT_TYPE, SecurityInfo As SECURITY_INFORMATION, ByRef pSidOwner As IntPtr, ByRef
    pSidGroup As IntPtr, ByRef pDacl As IntPtr, _
   ByRef pSacl As IntPtr, ByRef pSecurityDescriptor As IntPtr) As UInteger
    End Function

    Private Enum ACCESS_MASK As UInteger
        DELETE = &H10000
        READ_CONTROL = &H20000
```

```
        WRITE_DAC = &H40000
        WRITE_OWNER = &H80000
        SYNCHRONIZE = &H100000

        STANDARD_RIGHTS_REQUIRED = &HF0000

        STANDARD_RIGHTS_READ = &H20000
        STANDARD_RIGHTS_WRITE = &H20000
        STANDARD_RIGHTS_EXECUTE = &H20000

        STANDARD_RIGHTS_ALL = &H1F0000

        SPECIFIC_RIGHTS_ALL = &HFFFF

        ACCESS_SYSTEM_SECURITY = &H1000000

        MAXIMUM_ALLOWED = &H2000000

        GENERIC_READ = &H80000000UI
        GENERIC_WRITE = &H40000000
        GENERIC_EXECUTE = &H20000000
        GENERIC_ALL = &H10000000

        DESKTOP_READOBJECTS = &H1
        DESKTOP_CREATEWINDOW = &H2
        DESKTOP_CREATEMENU = &H4
        DESKTOP_HOOKCONTROL = &H8
        DESKTOP_JOURNALRECORD = &H10
        DESKTOP_JOURNALPLAYBACK = &H20
        DESKTOP_ENUMERATE = &H40
        DESKTOP_WRITEOBJECTS = &H80
        DESKTOP_SWITCHDESKTOP = &H100

        WINSTA_ENUMDESKTOPS = &H1
        WINSTA_READATTRIBUTES = &H2
        WINSTA_ACCESSCLIPBOARD = &H4
        WINSTA_CREATEDESKTOP = &H8
        WINSTA_WRITEATTRIBUTES = &H10
        WINSTA_ACCESSGLOBALATOMS = &H20
        WINSTA_EXITWINDOWS = &H40
        WINSTA_ENUMERATE = &H100
        WINSTA_READSCREEN = &H200

        WINSTA_ALL_ACCESS = &H37F
    End Enum

    <Flags()> _
    Private Enum SECURITY_INFORMATION As UInteger
        OWNER_SECURITY_INFORMATION = &H1
        GROUP_SECURITY_INFORMATION = &H2
        DACL_SECURITY_INFORMATION = &H4
        SACL_SECURITY_INFORMATION = &H8
        UNPROTECTED_SACL_SECURITY_INFORMATION = &H10000000
        UNPROTECTED_DACL_SECURITY_INFORMATION = &H20000000
        PROTECTED_SACL_SECURITY_INFORMATION = &H40000000
        PROTECTED_DACL_SECURITY_INFORMATION = &H80000000UI
    End Enum
```

```
Private Enum SE_OBJECT_TYPE
    SE_UNKNOWN_OBJECT_TYPE = 0
    SE_FILE_OBJECT
    SE_SERVICE
    SE_PRINTER
    SE_REGISTRY_KEY
    SE_LMSHARE
    SE_KERNEL_OBJECT
    SE_WINDOW_OBJECT
    SE_DS_OBJECT
    SE_DS_OBJECT_ALL
    SE_PROVIDER_DEFINED_OBJECT
    SE_WMIGUID_OBJECT
    SE_REGISTRY_WOW64_32KEY
End Enum

Public Sub New()
End Sub


' In the Init function, register for HttpApplication
' events by adding your handlers.
Public Sub Init(ByVal application As HttpApplication) _
        Implements IHttpModule.Init
    AddHandler application.BeginRequest, _
        AddressOf Me.Application_BeginRequest
    AddHandler application.EndRequest, _
        AddressOf Me.Application_EndRequest
End Sub

Private Sub Application_BeginRequest(ByVal source As Object, _
         ByVal e As EventArgs)
    ' Create HttpApplication and HttpContext objects to access
    ' request and response properties.
    Dim application As HttpApplication = DirectCast(source, HttpApplication)
    Dim context As HttpContext = application.Context
    Dim Status As String = "Not Authenticated"
    Dim username as String
    Dim filePath As String = context.Request.FilePath
    Dim fileExtension As String = VirtualPathUtility.GetExtension(filePath)
    If Not HttpContext.Current.Request.Cookies(FormsAuthentication.FormsCookieName) Is
    Nothing Then
        Dim ticket As FormsAuthenticationTicket =
        FormsAuthentication.Decrypt(HttpContext.Current.Request.Cookies(FormsAuthenticatio
        n.FormsCookieName).Value)
        username = ConfigurationSettings.AppSettings("Domain") & "\" &
        ticket.Name.ToString
        Dim Path As [String] = context.Request.PhysicalPath
        Dim pSidOwner As IntPtr, pSidGroup As IntPtr, pDacl As IntPtr, pSacl As IntPtr,
        pSecurityDescriptor As IntPtr
        Dim mask As New ACCESS_MASK()
        Dim ret As UInteger = GetNamedSecurityInfo(Path, SE_OBJECT_TYPE.SE_FILE_OBJECT,
        SECURITY_INFORMATION.DACL_SECURITY_INFORMATION, pSidOwner, pSidGroup, pDacl,
        pSacl, pSecurityDescriptor)
        Dim t As New TRUSTEE()
```

```
            t.TrusteeForm = TRUSTEE_FORM.TRUSTEE_IS_NAME
            t.TrusteeType = TRUSTEE_TYPE.TRUSTEE_IS_USER
            t.ptstrName = username
            ret = GetEffectiveRightsFromAcl(pDacl, t, mask)

            If (mask And ACCESS_MASK.READ_CONTROL) = ACCESS_MASK.READ_CONTROL Then
'                Status = username & " is allowed to read."
            Else
                context.Response.Redirect(ConfigurationSettings.AppSettings("NotAuthPage"))
'                Status = username & " is NOT allowed to read!"
            End If
        else
            context.Response.Redirect(ConfigurationSettings.AppSettings("LoginPage"))
        End If

'         context.Response.Write("<h1><font color=red>" & Status & "</font></h1><hr>")
    End Sub

    Private Sub Application_EndRequest(ByVal source As Object, ByVal e As EventArgs)

    End Sub

    Public Sub Dispose() Implements System.Web.IHttpModule.Dispose
    End Sub

End Class
```