KillTest

Higher Quality, Better Service!

Q&A

http://www.killtest.com

We offer free update service for one year.

Exam : **070-330**

Title : Implementing Security for Applications with Microsoft Visual Basic .NET

Version : DEMO

1. You are an application developer for your company. You develop an application that uses an external class library. You run the Permissions View tool on the class library and receive the following output.Microsoft (R) .NET Framework Permission Request Viewer. Version 1.1.4322.573Copyright (C) Microsoft Corporation 1998-2002. All rights reserved.minimal permission set: <PermissionSet class="System.Security.PermissionSet" version="1"><IPermission class="System.Security.Permissions.ReflectionPermission,

mscorlib, Version=1.0.5000.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"version="1" Flags="ReflectionEmit"/>

<IPermission class="System.Security.Permissions.SecurityPermission, mscorlib, Version=1.0.5000.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Flags="SerializationFormatter"/>

</PermissionSet> optional permission set:<PermissionSet class="System.Security.PermissionSet"

version="1" Unrestricted="true"/> refused permission set:

Not specified You need to add corresponding attributes in your application. Which code segment should you use?

A. <Assembly: ReflectionPermission(SecurityAction.RequestRefuse, _ ReflectionEmit:=False), _
Assembly: SecurityPermission(SecurityAction.RequestRefuse, _ SerializationFormatter:=False), _
Assembly: PermissionSetAttribute(SecurityAction.RequestOptional, Unrestricted:=True)>

B. <Assembly: ReflectionPermission(SecurityAction.RequestMinimum, _ ReflectionEmit:=False), _
Assembly: SecurityPermission(SecurityAction.RequestRefuse, _ SerializationFormatter:=False), _
Assembly: PermissionSetAttribute(SecurityAction.RequestRefuse, Unrestricted:=True)>

C. <Assembly: ReflectionPermission(SecurityAction.RequestMinimum, _ ReflectionEmit:=False), _
Assembly: SecurityPermission(SecurityAction.RequestMinimum, _ SerializationFormatter:=False), _
Assembly: PermissionSetAttribute(SecurityAction.RequestOptional, Unrestricted:=True)>

D. <Assembly: ReflectionPermission(SecurityAction.RequestMinimum, _ ReflectionEmit:=True), _
Assembly: SecurityPermission(SecurityAction.RequestMinimum, _ SerializationFormatter:=True), _
Assembly: PermissionSetAttribute(SecurityAction.RequestOptional, Unrestricted:=True)>
Answer: D

2. You are an application developer for your company. You create a Web application that is used by all users in the company. The application is hosted on the intranet Web server, which is named WebServer. WebServer has IIS 5.0 installed. The Web application is configured to use Integrated Windows authentication. The Web.config file specifies that the authentication mode is set to Windows.

The application connects to a Microsoft SQL Server database named DataStore. The database is located on WebServer. The SQL Server computer is configured with SQL Server logins disabled. The database connection code is shown in the following code segment.

Dim myConnStr As String myConnStr = "Initial Catalog=""DataStore"";" myConnStr = myConnStr & "Data Source=localhost;Integrated Security=SSPI;" Dim myConn As New SqlConnection(myConnStr) Dim myInsert As String

myInsert = "INSERT INTO Customer (CustomerID, Name) Values('123', 'John Doe')" Dim myCmd As New SqlCommand(myInsert) myCmd.Connection=myConn myConn.Open() myCmd.ExecuteNonQuery() myCmd.Connection.Close() When you run the application by using Microsoft Internet Explorer, you receive an error message that reads in part: "Login failed for user WebServer\ASPNET." You need to ensure that the application can run successfully without prompting the user for a user name and password. What should you do?

A. Change the authentication mode in IIS to basic authentication. Update the connection string.

B. Change the authentication mode in IIS to Anonymous and supply a login ID and password for a SQL Server login account that has access to the database. Update the connection string.

C. Enable Integrated Windows authentication in Internet Explorer.

D. Enable impersonation in the Web.config file.
Answer: D

3. You are an application developer for your company. You are developing a Windows Forms application. You deploy a

supporting assembly named MyAssembly.dll to the global assembly cache. During testing, you discover that theapplication is prevented from accessing MyAssembly.dll. You need to ensure that the application can access MyAssembly.dll. What should you do?

A. Digitally sign the application by using a digital certificate.

B. Run the caspol.exe -s on command from the command line.

C. Run the Assembly Linker to link MyAssembly.dll to the application.

D. Modify the security policy to grant the application the FullTrust permission.
Answer: D

4. You are an application developer for your company. You maintain a Windows Forms application. Data entry logic for the application is enforced by the user interface layer. The application contains assemblies that communicate data changes to the database. The application also contains assemblies that implement business logic. You create a new assembly named NewAssembly, which is called from the user interface. Values are passed to NewAssembly, which performs calculations by using the data. NewAssembly calls a separate assembly to store the resulting data in a database. You need to perform unit testing on the application to identify security vulnerabilities caused by unanticipated use of the application. Which two actions should you perform? (Each correct
Answer presents part of the solution. Choose two.)

A. Test the application by calling NewAssembly directly.

B. Test the application to verify whether it performs to the original functional specifications.

C. Test the application by using a domain administrator account.

D. Test the application by using the account of a user who should not have access to the application.
Answer: AD

5. You are an application developer for your company. You are testing an application that was developed by another developer. The application maintains its own list of authorized users. Each user is assigned a security level of 1, 2, or 3. When a new user account is created, the security level for that user is entered into a text box. The new user account information is saved in a Microsoft SQL Server table by using a stored procedure. You verify that user accounts that have any of the three security levels can perform only the intended actions within the application. You need to identify any security vulnerabilities in the portion of the application that creates new user accounts. What should you do first?

A. Use SQL Query Analyzer to create a new user account that has a security level of 2. Test the application to see if the new user account can log on to the application.

B. Create a new user account that has a security level other than 1, 2, or 3. Test the application to see what the new user account can do.

C. Use Osql.exe to call the stored procedure and create a new user account that has a security level of 3. Test the application to see what the new user account can do.

D. Create a new user account that has a security level of 3. Test the application to see what the new user account can do.
Answer: B

6. You are an application developer for your company. The company maintains an internal, self-signed certification authority (CA). You are releasing a new internal Windows Forms application. A written company policy prohibits internal applications from running on client computers unless the identity and integrity of those applications can be proven. The Microsoft .NET Framework on all client computers is configured to enforce this restriction. You need to ensure that your application will run when installed on all client computers. Your solution must not require any financial expenditure. What should you do?

A. Use a software publisher certificate issued by the internal CA to sign the application assemblies.

B. Use a software publisher certificate issued by a third-party commercial CA to sign the application assemblies.

C. Run the Certificate Creation tool and the Software Publisher Certificate Test tool before distributing the application to client computers.

D. Distribute the application as an e-mail attachment. Digitally sign the e-mail message before sending it to all company users.
Answer: A

7. You are an application developer for your company. You create an ASP.NET Web application. The application allows customers to select items for purchase. During the active session of a customer, data about the quantity and price of items selected by the customer is stored in a cookie on the client computer. You need to test the application for security vulnerabilities. What should you do?

A. Test the application by using a browser that has cookies disabled.

B. Test the application by selecting 150 items for purchase.

C. Test the application by using a cookie that you create in a text editor.

D. Test the application by using the five most common Internet browsers.
Answer: C

8. You are an application developer for your company. You are developing an application that can be extended by usingcustom components. The application uses reflection to dynamically load and invoke these custom components. In somecases, custom components will originate from a source that is not fully trusted, such as the Internet. You need toprogrammatically restrict the code access security policy under which custom components run so that customcomponents do not run with an elevated permission grant. What are two possible ways to achieve this goal? (Eachcorrect
Answer presents a complete solution. Choose two.)

A. Create a new application domain and set the security policy level. Run custom components in this application domain.

B. Use permission class operations to modify the security policy.

C. Implement custom permission classes to protect custom component resources.

D. Programmatically modify the machine-level security policy file after loading a custom component.
Answer: AB

9. You are an application developer for your company, which is named Humongous Insurance. You are developing anapplication to manage medical insurance claims. The application includes a serviced component named ClaimRecord.The business rules implemented by the application allow only those users who are members of theHumongousInsurance\ClaimsProcessor domain group to access the ClaimRecord component. You apply attributes to the ClaimRecord component to enable role-based security. You use the following assembly-levelattribute to add a role named ClaimsProcessor to the COM+ application that hosts the ClaimRecord component.

<Assembly: SecurityRole("ClaimsProcessor")> You deploy the ClaimRecord component to your staging server. You log on to the application by using a user accountthat is a member of the

HumongousInsurance\ClaimsProcessor domain group. When your application attempts to access the ClaimRecord component, an UnauthorizedAccessException exception is thrown.You need to modify the ClaimRecord component or reconfigure the COM+ application so that access is granted.You need to achieve this goal without compromising the security requirement of the ClaimRecord component.What should you do?

A. Replace the assembly-level attribute with the following attribute. <Assembly: SecurityRole("ClaimsProcessor", SetEveryoneAccess:=True)>

B. Replace the assembly-level attribute with the following attribute. <Assembly: SecurityRole("HumongousInsurance\ClaimsProcessor")>

C. Add the SuppressUnmanagedCodeSecurity attribute to the ClaimRecord component.

D. Using the Component Services tool, add the HumongousInsurance\ClaimsProcessor domain group to the COM+ClaimsProcessor role.

Answer: D

10. You are an application developer for your company. The company runs an e-commerce Web site. Users log on to theWeb site by using a password. Passwords are stored in a text file. The following code segment prepares the passwordsfor storage. Function HashPassword(ByVal Pwd As String) As StringReturn FormsAuthentication.HashPasswordForStoringInConfigFile(Pwd, "SHA1")End FunctionUsers of the Web site are creating passwords that are easily cracked by dictionary attacks.You need to decrease the likelihood that a dictionary attack will succeed if the password file is stolen, without restrictingthe passwords that users can create. What should you do?

A. Create a dictionary file that contains common words. Write additional code to reject passwords that match the entriesin the dictionary.

B. Apply a more restrictive discretionary access control list (DACL) to the password storage file.

C. Replace the HashPassword function with the following code segment. Function HashPassword(ByVal Pwd As String)As String Return FormsAuthentication.HashPasswordForStoringInConfigFile(Pwd, "MD5")End Function

D. Replace the HashPassword function with the following code segment. Function HashPassword(ByVal Pwd As String) As String Dim Rng As New RNGCryptoServiceProvider Dim Salt(16) As Byte Rng.GetBytes(Salt) Dim saltstr As String = Convert.ToBase64String(Salt) Return saltstr &FormsAuthentication.HashPasswordForStoringInConfigFile(_ saltstr & Pwd, "SHA1")End Function

E. Replace the HashPassword function with the following code segment. Function HashPassword(ByVal Pwd As String) As String Dim Hash As Integer = 0 Dim Enc As New UnicodeEncoding Dim HashData As Byte() = Enc.GetBytes(Pwd) Dim i As Integer For i = 0 To HashData.Length Step 2 Hash = Hash Xor

(HashData(i) Or (HashData(i + 1) << 8)) Next Return Hash.ToString()End Function
Answer: D