

Exam 70-340 study material

Made available by Examsexpert.com



Free 70-340 Exam Preparation Questions

Exam 70-340: Implementing Security for Applications with Microsoft Visual C# .NET

Question: 1

You are an application developer for Company.com. You are conducting a code review of an assembly written by another developer. The assembly is named MyAssembly.exe. The assembly is for an application that accesses data in a Microsoft SQL Server database. All users of the application have access to the database by using their Microsoft Windows user accounts. The assembly contains the following code segment.

```
string userid; string password; userid = "sa";password = ""; SqlConnection sqlConnection = new
SqlConnection();string connectionString; connectionString = "data source=myServer"; connectionString
+= ";initial catalog=myDatabase"; connectionString += ";user id=" + userid; connectionString +=
";password=" + password; sqlConnection.ConnectionString = connectionString; sqlConnection.Open();
```

You need to improve the security of the code segment.

What should you do?

- A. Replace the code segment with the following code segment. `SqlConnection sqlConnection = new SqlConnection(); string connectionString; connectionString = "data source=myServer"; connectionString += ";Integrated Security=SSPI"; connectionString += "initial catalog=myDatabase"; sqlConnection.ConnectionString = connectionString; sqlConnection.Open();`
- B. Replace the code segment with the following code segment. `SqlConnection sqlConnection = new SqlConnection(); string connectionString; connectionString = "data source=myserver;initial catalog=myDatabase;user id=sa;password sqlConnection.ConnectionString = connectionString;`
`sqlConnection.Open();`
- C. Run the `caspol.exe -resolvperm MyAssembly.exe` command from the command line.
- D. Run the `permview /decl MyAssembly.exe` command from the command line.

Answer: A**Explanation**

Never use the SQL default administrative account 'SA' and a blank password "", for any sort of access. This account has all access to all databases regardless of who or what created it as well as can be used to take complete control of the machine and even the network. SQL has hundreds of extended stored procedures (XP_???) of them xp_cmd can be use to elevate permissions well beyond what is needed and be used to compromise almost every aspect of the system and the network.

Security Recommended Practices

Microsoft recommends the following practices to help you protect your data and applications from malicious users and accidental user actions. Notification Services Security Practices

- Run the NS\$*instance_name* service under a weak domain or local account. Do not use the LocalSystem or NetworkService service account or any account in the **Administrators** group. However, if you are using a delivery protocol that requires the account that the service runs under to have additional privileges, you must use higher privileges. For example, sending notifications using an Internet Information Services (IIS) SMTP server requires the account under which the service runs to be a member of the local Administrators group.
- Ensure that the password used by the service account is a strong password. For more information about strong passwords, see "Creating Strong Passwords" in the Microsoft Windows documentation.
- Ensure that all code run by the NS\$*instance_name* service, such as custom event providers, content formatters, and protocols, is from a trusted source. Notification Services assumes that code listed in the application definition file (ADF) comes from a trusted source.
- Secure all folders containing configuration files or application data. For more information about securing files and folders, see File and Folder Security. SQL Server Security Practices
- When installing SQL Server, never allow a blank sa password, even if you select the integrated security mode. This guarantees that if the security mode changes to mixed mode, the sa account will still have a password.
- Use Windows Authentication whenever possible. Windows Authentication provides advanced

security features, such as policies for password length, complexity, and expiration. Note that if the `NS$instance_name` service uses a SQL Server user name and password to connect to SQL Server, this user name and password are encrypted and stored in the registry.

- If you use SQL Server Authentication, use strong passwords for the SQL Server login accounts and change the passwords periodically.
 - Do not grant unnecessary permissions to the public role in each database. The public role is a special database role to which every database user belongs, and cannot be dropped from the database. Notification Services does not use the public role.
 - Do not grant database access to the guest user account. The guest user account allows a SQL Server login account that does not have a database user account to access a database.
 - Consider encrypting the database files using NTFS file encryption. This can decrease performance, so you must weigh optimal performance against file security. Network Communications Security Practices
 - To reduce the possibility of intruders viewing data as it is being transferred between Notification Services and the database, use encrypted communication between client applications and SQL Server. For more information, see "Using Encryption Methods" in SQL Server Books Online.
 - If you are using an HTTP protocol to post data to a Web server, and if the Web server supports SSL, post the notification using an address that starts with `https://`. This form of address encrypts the data that is sent to the Web server. Physical Security Practices
- Ensure that your servers are located in an area that is adequately secured. If a malicious user can physically access the server, the server is not secure. Database Security
- One of the most common scenarios for a distributed application involves reading and writing data on a remote database. The dilemma that arises is how to do so securely while maintaining application scalability. Where you choose to manage security in your application will greatly impact, either negatively or positively, the scalability of your application. To achieve scalability using database connection pooling foregoes having the database manage security. This is because database connection pooling requires the connection string be identical to pool connections. Therefore, you must manage security elsewhere. If you must track database operations on per user basis, consider adding a parameter for user identity to each operation and manually log user actions in the database. Following the advice above, another issue is how to store the database connection string, which typically contains security credentials, so multiple users can access it without compromising security. Most sample applications demonstrate storing the connection string in the `Web.config` or `global.asax` files. However, because these files are plain text files that have limited security, it is not the best location for storing this information. Should an intruder compromise your Web server's security, these files would be easily accessible. Here are just a few alternatives:
- If using the `Web.config` file, store the connection string encrypted and then decrypt the connection string in your application code when needed.
 - Build a COM+ application using the `ServiceComponent Class` and store the connection string in the construct string for that component. When storing sensitive information in the constructor string, you should verify the following:
 - Only the appropriate users/groups belong to the Reader role of the System Package. However, you must carefully manage COM+ to prevent it from being unable to read its own configuration.
 - You have controlled and audited access to the `%windows%\Registration` folder, where the COM+ configuration database (RegDB) stores its files. For more information, see `ServiceComponent Class`.
 - Use integrated security to make a trusted connection with SQL Server. This makes it possible for you to use a connection string that eliminates the need for storing a password in the connection string, such as: `"Data Source=mySqlServer;Integrated Security=SSPI;Initial Catalog=myDB"` There are some drawbacks to using integrated security, most of which you can overcome. Because integrated security requires a Windows account, it defeats connection pooling if you impersonate each authenticated principal using an individual Windows account. However, if you instead impersonate a limited number of Windows accounts, with each account representing a particular role, you can overcome this drawback. Each Windows account must be a domain account with IIS and SQL Server in the same or trusted domains. Alternatively, you can create identical (including passwords) Windows accounts on each machine. After a typical installation, the default security authentication mode is Windows Authentication for SQL Server 2000, which is different from SQL Server 7.0. In SQL Server

7.0, the default authentication mode is Mixed (Windows Authentication Mode and SQL Server Authentication). Windows Authentication is a better security method because of the additional security features it provides, such as secure validation and encryption of passwords, password expiration and auditing. For more information, see Authentication Modes. If you configure SQL Server to use Windows Authentication, you could create one Windows account for read-only operations and another Windows account for read/write operations. You then map each Windows account to a SQL Server login and establish the desired permissions. Using application logic, you then determine which Windows account to impersonate when performing database operations. In SQL Server, you can add any Windows user account as a

member of a fixed database role. Each member gains the permissions applied to the fixed database role. For more information, see Managing Permissions. For SQL Server 7.0, integrated security does not work with SQL Server's TCP/IP network library, but uses the named pipes network library instead. As an added security measure, the ConnectionString property of the SqlConnection object does not persist or return the full connection string by default. To do so, you must set Persist SecurityInfo to true.

Question: 2

You are an application developer for your company. You are conducting a code review of an application that updates a Microsoft SQL Server database named Payroll. This database is used by other applications. The application contains the following code segment.

```
public void calculateAndStore(SqlConnection conn, string ID, string bonus, string salary) {  
    salary=Convert.ToDecimal(Convert.ToDecimal(salary)*1.05m);  
    bonus=Convert.ToString(Convert.ToDecimal(salary)*0.05m); if (ID.Length==0) throw new  
    ApplicationException("Error - Empty ID"); string newID="ID-" + ID; string strUpdate = "UPDATE Payroll  
    SET EmployeeID = " + newID + ", Bonus = " + bonus + ", Salary = " + salary + " " + "WHERE  
    EmployeeID=" + ID + """; SqlCommand cmd=new SqlCommand(strUpdate,conn);  
    cmd.Connection.Open(); cmd.ExecuteNonQuery(); } The values in the string variables named ID,  
    Bonus, and Salary are contained in the Payroll database. The purpose of the code segment is to  
    calculate new values for ID, Bonus, and Salary, and to update those values in the Payroll database. You  
    need to improve the security of this application. What should you do?
```

- A. Validate that the Salary value is within the range for the data type in the SQL Server database.
- B. Validate the contents of the ID value before updating it in the SQL Server database.
- C. Validate the length of the ID value before updating it in the SQL Server database.
- D. Enclose the body of the function within a try-catch block.

Answer: B

Question: 3

You are an application developer for Company.com. You are developing a client The application uses an unmanaged component to retrieve data from another application, and your application uses that data as part of a SQL query. In the application code, you use a variable named externalobject to refer to the unmanaged component. A variable named calcval contains an integer value that is calculated by your application. A SqlCommand object named sqlcmd is already defined and associated with an open ADO.NET connection to the SQL Server database. The application contains the following code segment.

```
string myquery; myquery = "INSERT INTO DataStore (ExternalID, CalcValue)"; myquery += " VALUES(" +  
    externalobject.LegacyData + ","; myquery += calcval.ToString() + ")"; sqlcmd.CommandText =  
    myquery;  
    sqlcmd.ExecuteNonQuery();
```

You need to improve the security of this code segment.

What should you do?

- A. Place the code segment within a try-catch block.
- B. In the code segment, ensure that the value of externalobject.LegacyData meets the length and type requirements of the SQL Server table.

- C. Validate the externalobject.LegacyData contains only expected data and no additional SQL statements.
- D. Copy the contents of externalobject.LegacyData into a string variable, and append the string variable to the SQL statement.

Answer: C

Explanation

It is best for the approach that you must validate and cleanse data before it is used and/or store. Rule number two is: *data must be validated as it crosses the boundary between untrusted and trusted environments*. By definition, trusted data is data you or an entity you explicitly trust has complete control over; untrusted data refers to everything else. In short, any data submitted by a user is initially untrusted data. When it comes to SQL statements, all dynamic SQL is bad, and parameterized stored procedures must be used. Dynamic SQL can be easily compromised and used for SQL injection attacks. All relational databases—including SQL Server, Oracle, IBM DB2, and MySQL—are susceptible to SQL injection attacks. You can buy products that protect your system from SQL injection, but for most businesses, the defense against SQL-injection attack must be code-based. The opening for SQL-injection attacks comes primarily through Web applications that combine user input with dynamic SQL to form SQL commands that the application sends to the database. Here are four important steps you can take to protect your Web applications from SQL-injection attacks. In addition to the following tips, the Microsoft Patterns and Practices Library that I highlighted last month provides advice about securing your data-access applications.

4. Principle of Least Privilege The account an application uses to connect to the database should have only the privileges that application requires. The security permissions that an intruder gains from a compromised application define the harm that the intruder can inflict. Applications shouldn't connect as sa or with the Administrator account. Instead, the account should have permissions to access only the database objects it needs.

3. Validate All Input If an input field should contain numeric data, then verify that users enter only numbers. If character data is acceptable, check for unexpected characters. Make sure your application looks for characters such as semicolons, equals signs, double dashes, brackets, and SQL keywords. The .NET Framework provides regular expressions that enable complex pattern matching, a good way to test user input. Limiting the length of accepted user input is also a good idea. Validating your input might seem obvious, but many applications are vulnerable to SQL-injection attacks because intruders can use the openings that Web applications offer.

2. Avoid Dynamic SQL Dynamic SQL is a great tool for performing ad hoc queries, but combining dynamic SQL with user input creates exposure that makes SQL-injection attacks possible. Replacing dynamic SQL with prepared SQL or stored procedures is feasible in most applications. Prepared SQL and stored procedures accept user input as parameter data rather than as SQL commands, thus limiting what an intruder can do. Of course, replacing dynamic SQL with a stored procedure won't help you if you use the user input to build dynamic SQL statements in your stored procedures. In that case, the dynamic SQL that the user input creates will still be corrupted, and your database will still be in danger of SQL-injection attack.

1. Use Double Quotes Replace all the single quotes that your users' input contains with double quotes. This simple precaution will go a long way toward warding off SQL-injection attacks. Single quotes often terminate SQL expressions and give the input more power than is necessary. Replacing the single quotes with double quotes will cause many SQL injection attacks to fail.

Question: 4

You are an application developer for Company.com. You are examining 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?

- A. Use SQL Query Analyzer to create a new user account that has a security level of 2. Exam 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. Exam the application to see what the new user account can do.
- C. Use Osq.exe to call the stored procedure and create a new user account that has a security level of 3. Exam the application to see what the new user account can do.
- D. Create a new user account that has a security level of 3. Exam the application to see what the new user account can do.

Answer: B

Explanation

Security testing is about validating your application's security services and identifying potential security flaws. This section contains important testing recommendations for verifying that you have created a securable application. Since attackers have no standard method of breaking into things, there are no standard methods of conducting security testing. Also, there are few tools available at this time to test security aspects thoroughly. Since a functional bug in an application can also represent a potential security flaw, you need to conduct functional testing prior to conducting security testing. It is important to note that security testing will not prove conclusively that an application is secure. Instead, it serves only to validate the effectiveness of instituted countermeasures, which were chosen based upon presumptions that were made during the threat analysis phase. Provided below are some suggestions for testing the securability of your application. There are some security issues you should be aware of when you test your smart documents. These security measures, described in the Security section, are in place to provide security for Microsoft® Office 2003 users. However, during testing, you may want to disable the XML expansion pack security check, if possible, or you may want to create a test environment that meets the security requirements of your users. The following topics provide additional information about security within a development and testing environment:

- Disabling the XML Expansion Pack Security Check
- Digital Code Signing for Testing Purposes
- Creating a Digital Certificate for Testing Purposes
- Delay Signing a Smart Document
- Assembly Testing a Signed XML Expansion Pack

Test for Buffer Overflows

One of the first security bugs exploited in computer history was a buffer overflow. Buffer overflows continue to be one of the most dangerous and most commonly occurring weaknesses. Attempts to exploit this type of vulnerability can result in problems ranging from crashing the application to an attacker inserting and executing malignant code in the application process. When writing data to buffers, it is imperative that developers not write more to the buffer than it can possibly hold. If the amount of data being written exceeds the buffer space that has been allocated, a buffer overflow occurs. When a buffer overflow occurs, data is written into parts of memory that may be allocated for other purposes. A worst-case scenario is when the buffer overflow contains malicious code that is then executed. Buffer overflows account for a large percentage of security vulnerabilities.

Conduct source code security reviews

Depending upon the sensitivity of the application in question, it might be prudent to conduct a security audit of the application source code. A source code audit should not be confused with a code review. The purpose of a standard code review is to identify general code defects that affect the functionality of the code. The purpose of a source code security review is to identify security flaws, intentional or otherwise. Such a review would be especially warranted when developing applications that handle financial transactions or provide for public safety.

Validate contingency plans

There will always be a potential that an application's security defenses can be breached and it is only prudent that contingency plans are in place and validated. What steps will be taken if a virus is detected

on your application server or in your data center? When security is thwarted, reactions must occur rapidly to prevent further damage. Find out if your contingency plans will work before they must be battle-tested.

Attack your application

Testers are accustomed to tormenting applications in an attempt to make them fail. Hacking your own application is a similar, but more focused, process. When attempting to attack your application, you should be looking for exploitable flaws that represent a weak spot in your application's defenses.

Question: 5

You are an application developer for your company. You create an ASP.NET Web application. The application accesses data that is stored in a Microsoft SQL Server database named EmployeeData. EmployeeData contains a table named Payroll that contains payroll information including a column named Salary and a column named EmployeeName. The Salary column uses a decimal data type. The EmployeeName column uses an nvarchar data type with a length of 200.

The application uses the following code to connect to the EmployeeData database and update payroll records. SqlConnection myConn = new SqlConnection(myConnStr); string myUpdate = "UPDATE Payroll SET Salary = " + Salary.ToString()

```
    + " WHERE EmployeeName = " + EmployeeName + """; SqlCommand myCmd = new
SqlCommand(myUpdate); myCmd.Connection = myConn; myConn.Open();
myCmd.ExecuteNonQuery(); myCmd.Connection.Close();
```

You need to perform unit testing on the application to identify vulnerability to SQL injection attacks. What should you do?

- A. Enter employee names that are found in the database.
- B. Enter negative values for the salary.
- C. Enter a value such as ' OR 1=1 -- for an employee name.
- D. Enter user names that exceed 200 characters.

Answer: C

Question: 6

You are an application developer for your company. You create an ASP.NET Web application that is hosted on an intranet Web server named Server1. The application is configured to use Forms authentication. The application requires users to log on by using a user name and password. The application stores user names and passwords in a Microsoft SQL Server database that is located on a database server named Server2. The login pages for the application use SSL/TLS encryption. No other pages in the application use SSL/TLS. You need to test the application to find out if unauthorized users can view user name and password information.

What should you do?

- A. Access the application by using a user account that is a member of the SQL Server db_owner role.
- B. Access the login pages by using HTTP.
- C. Test that users who enter incorrect user name and password combinations are prevented from accessing the application.
- D. Capture and analyze the network traffic between Server1 and Server2

Answer: D

Question: 7

You are an application developer for your company. You are developing an application that will be distributed to partner companies that do business with your company. The partner companies must be able to verify the authenticity of the application's assemblies before they will install the application. You need to ensure that the assemblies meet the requirements of the partner companies. You want your solution to minimize the number of additional configuration steps required by the partner companies. What should you do?

- A. Use a certificate issued by your company's internal, self-signed certification authority (CA) to

- digitally sign the assemblies.
- B. Use a certificate issued by a third-party commercial certification authority (CA) to digitally sign the assemblies.
 - C. Run the PEVerify tool before distributing the application to partner companies.
 - D. Run the Software Publisher Certificate Test tool before distributing the application to partner companies.

Answer: B

Question: 8

You are an application developer for your company. You are conducting a code review of an application that was developed by another developer. The application declares variable named

perms. The value of this variable indicates which permissions a user has for the application. Each value represents a specific permission or set of permissions, and user groups are permitted to have specific permissions. The permissions and groups are shown in the following table. The application stores the current user's group name in a variable named group. Each user can belong to only one group. The application sets the value of perms as shown in the following code segment.

```
int perms = 4; if(group == "Editors") { perms -= 1; } if(group == "Reviewers") { perms -= 3; }
```

You need to improve the security of this code segment as much as possible while maintaining its functionality. You decide to replace the existing code segment.

Which code segment should you use?

- A.

```
int perms = 4; if(group != "Admins") { perms -= 1; } if(group != "Editors") { perms -= 2; } if(group != "Reviewers") { perms -= 1; }
```
- B.

```
int perms = 0; if(group == "Admins") { perms = 4; } if(group == "Editors") { perms = 3; } if(group == "Reviewers") { perms = 1; }
```
- C.

```
int perms = 1; if(group == "Admins") { perms += 3; } if(group == "Editors") { perms += 2; }
```
- D.

```
int perms = 1; switch(group) { case "Admins": perms = 4; break; case "Editors": perms = 3; break; }
```

Answer: B

Question: 9

You are an application developer for your company. You are developing an application that will be used to display the text associated with two particular file types. A file name extension of .nsa indicates that the file contains static text to be displayed. A file name extension of .nsax indicates that the file contains instruction codes for generating the text that should be displayed. These are the only two file types that the application recognizes. While testing the application, you discover that when a user attempts to open a file named File~1.nsa, the application displays the instruction codes contained in File.nsa instead of displaying the text that would be generated by those instruction codes. You need to prevent users from displaying the instruction codes contained in the .nsax files. What are two possible ways to achieve this goal? (Each correct answer presents a complete solution. Choose two.)

- A. Call the unmanaged GetLongPathName function for each file name before examining the file name extension.
- B. Reject all files that contain a tilde (~) in their names.
- C. Ask an administrator to disable the creation of short path names in the file system.
- D. Before choosing how to display a file, open the file and examine the contents to see if the file

contains text or instruction codes.

Answer: A, C

Question: 10

You are an application developer for your company. You are developing an application that includes administrative features that could destroy important information if used incorrectly. You need to ensure that only members of the Administrators group can use or discover the administrative features. You need to achieve this goal while minimizing the impact on users. Which two actions should you perform? (Each correct answer presents part of the solution. Choose two.)

- A. Disable the menu choices for all administrative features if the user is not a member of the Administrators group.
- B. Remove the menu choices for all administrative features if the user is not a member of the Administrators group.
- C. Throw an exception at the start of each administrative feature if the user is not a member of the Administrators group.
- D. Require users to provide an administrator password before each execution of administrative features in the application.

Answer: B, C

Question: 11

You are an application developer for your company. You are developing an application that stores and retrieves data in a Microsoft SQL Server database. The application accepts input from the user and stores the input in a variable named strInput. The input is to be saved in a SQL Server nchar column that has a length of 10. The application calls a stored procedure to save the input, and the stored procedure stores the input in a parameter named @varInput.

You need to ensure that input by the user can be stored in the SQL Server column without causing an exception. Your solution must ensure that either the entire user input is saved, or none of the user input is saved. Which two actions should you perform? (Each correct answer presents part of the solution. Choose two.)

- A. In the application, reject the input if the strInput.Length value is greater than 10.
- B. In the application, pass only strInput.Chars(9) to the stored procedure.
- C. In the application, use a regular expression to remove all non-alphanumeric characters.
- D. In the stored procedure, reject the input if the length of @varInput is greater than 10.
- E. In the stored procedure, save only the result of LEFT(@varInput,10) to the database column.
- F. In the stored procedure, save the result of REPLACE(@varInput,"@-]", "") to the database column.

Answer: A, D

Question: 12

You are an application developer for your company. You are developing a Windows Service application. Your user account is a member of only the Users group on your computer. A written company policy states that developers are not allowed to log on by using an account that has more authority than is needed. You need to develop and debug the application.

What should you do?

- A. A.Modify the application to run the Microsoft Visual Studio .NET debugger with the FullTrust permission by using code access security policy.
- B. Set the discretionary access control list (DACL) permissions on the executable file for the application to grant your user account Full Control permission for the executable file.
- C. Create a user account that is a member of the Debugger Users group. Use this account for

debugging

- D. Start the development environment from the command line by running the runas command and specifying an account in the Administrators group.

Answer: C

Question: 13

You are an application developer for your company. You are reviewing the security for a console application that was written by another developer. The application uses impersonation to run as a member of the Administrators group. The following code segment is the only code that deals with security in the application.

```
RegistryKey key = Registry.CurrentUser.CreateSubKey("Name");  
key.SetValue("Name", "Tester");
```

You need to improve the security of the application.

What should you do?

- A. Change the application to run as the interactive user.
- B. Run the application from the command line by using the runas command and specify the Administrator account.
- C. Change the application to use code access security.
- D. Change the application to write to the HKEY_LOCAL_MACHINE hive.

Answer: A

Question: 14

You are an application developer for your company, which is named Contoso, Ltd. You are developing an application that stores configuration data in a file named C:\Contoso\Persistence.config. This file is the only file your application will access. The design document for the application specifies the following two requirements: All authenticated users are allowed to view the contents of the configuration data file. Only members of a group named Managers are allowed to modify the data in the configurationdata file. You need to ensure that the file can be accessed according to these requirements.

Which two actions should you perform? (Each correct answer presents part of the solution. Choose two.)

- A. Apply a discretionary access control list (DACL) entry on the file. Use the DACL to grant Read permission to all authenticated users, and to grant Write permission to the Managers group.
- B. Apply a discretionary access control list (DACL) entry on the file. Use the DACL to grant Read permission to the Everyone group, and to grant Write permission to the Managers group.
- C. Add the following code segment to the application before accessing the file. { WindowsPrincipal wp=new WindowsPrincipal(WindowsIdentity.GetCurrent()); bool bManager=wp.IsInRole("Managers"); FileIOPermission fp; string configFile="C:\\contoso\\persistence.config"; if (bManager) fp=new FileIOPermission(FileIOPermissionAccess.AllAccess,configFile); else fp=new FileIOPermission(FileIOPermissionAccess.Read,configFile); fp.PermitOnly(); }
- D. Add the following code segment to the application before accessing the file. { WindowsPrincipal wp=new WindowsPrincipal(WindowsIdentity.GetCurrent()); if (wp.IsInRole("Managers")){ WindowsIdentity anon=WindowsIdentity.GetAnonymous(); anon.Impersonate(); } }
- E. Require users to enter a password that is shared among members of the Managers group whenever the application opens the file for writing.

Answer: A, C

Question: 15

You are an application developer for your company. You are developing a Windows Forms client application that will be used within the company to access data in a Microsoft SQL Server database. The application defines a SqlConnection object named sqlconn and a SqlCommand object named sqlcmd. The application also includes the following code segment. SqlDataReader sqlreader = sqlcmd.ExecuteReader(); You need to improve the security of this code segment. You decide to replace the existing code segment.

Which code segment should you use?

- A. `try { SqlDataReader sqlreader = sqlcmd.ExecuteReader(); } catch { MessageBox.Show("An error occurred while querying SQL Server."); }`
- B. `SqlDataReader sqlreader; sqlreader = sqlcmd.ExecuteReader(); sqlconn.Close();`
- C. `try { SqlDataReader sqlreader; sqlreader = sqlcmd.ExecuteReader(); } catch (Exception e) { MessageBox.Show(e.Message); }`
- D. `SqlDataReader sqlreader; sqlreader = sqlcmd.ExecuteReader(); if (! sqlreader.HasRows) { MessageBox.Show("An error occurred while querying SQL Server."); }`

Answer: A

Question: 16

You are an application developer for your company. Your development computer is named Dev1, and the computer has IIS 5.0 installed. You log on to Dev1 by using an account named User1. On Dev1, you develop an ASP.NET Web application for testing user authentication. Your project uses the default settings from Microsoft Visual Studio .NET 2003. Your Web application contains a form that includes a label control named Label1. The Machine.config file contains the following code segment.

```
<configuration> <system.web><authentication mode="None"> </authentication>
```

</system.web></configuration> The Web.config file contains the following code segment.

```
<configuration> <system.web><authentication mode="Windows"></authentication>
```

```
</system.web></configuration> The application contains the following code segment. Label1.Text = WindowsIdentity.GetCurrent().Name.ToString(); When you run the application, you discover that the text property of Label1 is set to Dev1\ASPNET. You need to ensure that the application sets the text property of Label1 to Dev1\User1.
```

What should you do?

- A. Replace the code segment in the Machine.config file with the following code segment.

```
<configuration> <system.web> <authentication mode="Windows"> </authentication> </system.web></configuration>
```
- B. Replace the code segment in the Web.config file with the following code segment.

```
<configuration> <system.web> <authentication mode="Windows"> </authentication> <identity impersonate="true" /> </system.web> </configuration>
```
- C. Replace the code segment in the application with the following code segment.

```
WindowsAccountType myIdentity = new WindowsAccountType(); Label1.Text = myIdentity.Normal.ToString();
```
- D. Replace the code segment in the application with the following code segment.

```
WindowsAccountType myIdentity = new WindowsAccountType(); Label1.Text = myIdentity.ToString();
```

Answer: B

Question: 17

You are an application developer for your company. The company uses code access security extensively in several applications that are installed on a server named AppServer1. You develop an application named Application1. Application1 requires a customized machine policy on AppServer1.

Another developer develops an application named Application2. Application1 and Application2 are installed on the same physical partition on AppServer1. The developer of Application2 makes a change to the machine-level security policy to support a change in Application2. After the change is implemented, users report that Application1 returns error messages when users try to run it. You need to ensure that Application1 can run successfully.

What should you do?

- A. Recover the previous machine policy level by using the Code Access Security Policy tool.
- B. Reset the user policy by using the Code Access Security Policy tool.
- C. Move Application1 to a different physical partition
- D. Install the Microsoft .NET Framework 1.0.

Answer: A

Question: 18

You are an application developer for your company. You are developing a three-tier Windows Forms application that will be used to manage confidential records. The business layer includes a remote object that is installed on an application server. The remote object is hosted in ASP.NET on the application server. IIS is configured to use Integrated Windows authentication, and ASP.NET is configured to use Windows authentication. All client computers and servers on the network support Kerberos authentication. The Windows Forms application communicates with the remote object by using a remoting proxy named myProxy. The remote object accesses a Microsoft SQL Server database. Permissions to database objects are granted based on the identity of the user. The remote object needs to run under the security context of the user. You need to write the code in the Windows Forms application that will configure the remoting proxy to have the credentials to use for authentication. Which code segment should you use?

- A. `IDictionary channelProperties; channelProperties = ChannelServices.GetChannelSinkProperties(myProxy); channelProperties["credentials"] = CredentialCache.DefaultCredentials;`
- B. `IDictionary channelProperties; NetworkCredential cred = new NetworkCredential(_userName, _psswd); channelProperties = ChannelServices.GetChannelSinkProperties(myProxy); channelProperties["credentials"] = cred;`
- C. `IDictionary channelProperties; channelProperties = ChannelServices.GetChannelSinkProperties(myProxy); channelProperties["credentials"] = Thread.CurrentPrincipal`
- D. `IDictionary channelProperties; channelProperties = ChannelServices.GetChannelSinkProperties(myProxy); channelProperties["credentials"] = Thread.CurrentPrincipal.Identity;`

Answer: A

Question: 19

You are an application developer for your company. You are conducting a code review of an e-commerce application. The application stores customer records, contact information, and credit card numbers in a customer database. You want to improve the security of the data. You need to write a function that prevents unauthorized users from reading database information while ensuring that the database can be read by the application for authorized customer access.

What should you do?

- A. Use the following function. `string PrepareString(string stringval, RSACryptoServiceProvider rsa) { byte[] data=Encoding.UTF8.GetBytes(stringval); byte[] retval=rsa.Encrypt(data, false); return Convert.ToBase64String(retval); }`
- B. Use the following function. `string PrepareString(string stringval, RSACryptoServiceProvider rsa) { byte[] data=Encoding.UTF8.GetBytes(stringval); byte[] retval=rsa.SignData(data,new SHA1CryptoServiceProvider()); return Convert.ToBase64String(retval); }`
- C. Write a function to encode all data by using the `Convert.ToBase64String` method.

D. Write a function to store a cryptographic hash of the data in the database.

Answer: A

Question: 20

You are an application developer for your company, which is named Humongous Insurance. You are developing an application 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 the HumongousInsurance\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-level attribute 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 account that 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

Question: 21

You are an application developer for your company. You are developing a multithreaded application. Some of the application's threads perform maintenance tasks in the application's database. These maintenance tasks are performed by dedicated assemblies, and the assemblies need to run under different security permissions. The other assemblies of the application must not have the different permissions. You need to ensure that the application's threads have the correct security permissions. You want to achieve this goal without negatively affecting response times for the application. What should you do?

- A. Configure the application to impersonate a user account that has the permissions required by the maintenance assemblies.
- B. Configure code access security policies so that the application has the permissions required by the maintenance assemblies.
- C. Create a separate application domain for the maintenance assemblies
- D. Start the maintenance assemblies as separate processes.

Answer: C

Question: 22

You are an application developer for your company, which is named Fourth Coffee. You are developing a Windows Forms application. Users will run your application from a Web folder on the intranet. The application stores configuration information in isolated storage. The application will read from the registry if it has the appropriate permission, but the application can run successfully without this permission.

You add the following attribute to the application. [assembly:

RegistryPermission(SecurityAction.RequestOptional, All=@"HKEY_CURRENT_USER\Software\Fourthcoffee\LastRun"])When you run the application from the intranet, a SecurityException exception is thrown. You need to modify attributes to indicate the application's exact permission requirements and correct the problem that is causing the SecurityException exception.

What should you do?

- A. Add the following attributes to the assembly. [assembly: IsolatedStorageFilePermission(SecurityAction.RequestMinimum, UsageAllowed=IsolatedStorageContainment.AssemblyIsolationByUser, UserQuota=9223372036854775807)] [assembly: UIPermission(SecurityAction.RequestMinimum)]
Make no change to the RegistryPermission attribute.
- B. Add the following attributes to the assembly. [assembly: IsolatedStorageFilePermission(SecurityAction.RequestMinimum, UsageAllowed=IsolatedStorageContainment.AssemblyIsolationByUser, UserQuota=9223372036854775807)] [assembly: UIPermission(SecurityAction.RequestMinimum)]
Replace the RegistryPermission attribute with the following attribute. [assembly: RegistryPermission(SecurityAction.RequestRefuse, All=@"HKEY_CURRENT_USER\Software\Fourthcoffee\LastRun")]
- C. Remove the RegistryPermission attribute and add the following attribute to the assembly. [assembly: PermissionSet(SecurityAction.RequestMinimum, Name="FullTrust")]
- D. Remove the RegistryPermission attribute and add no new attributes to the assembly.

Answer: A

Question: 23

You are an application developer for your company. You are modifying an existing communications application so that the application can be used on the Internet. You need to enhance the security of data when it is transmitted by using the application. You want to achieve this goal by using the minimum amount of development effort.

What should you do?

- A. Use HTTPS for all exchange of data.
- B. Encrypt the data by using the sender's public key and the recipient's private key.
- C. Encrypt the data by using the sender's private key and the recipient's public key.
- D. Create a random key to encrypt data for each exchange between the sender and recipient. Share the random key with the recipient. Encrypt subsequent data by using the shared random key.
- E. Create a random key to encrypt data for each exchange between the sender and recipient. Encrypt the random key by using the recipient's public key. Send the encrypted random key to the recipient. Encrypt subsequent data by using the shared random key.

Answer: A

Question: 24

You are an application developer for your company. Your team is developing a Windows Forms application. their roles in the company. The application includes the following method. static bool AuthenticateUser(string user, string password, out string[] roles) This method authenticates the user against a third-party data store. When authentication issuccessful, this method returns a value of true, and the string array named roles is updated tocontain the user's roles. You need to write the code that associates an authenticated user and the user's roles with the current security context.

Which code segment should you use?

- A. // PrincipalPermission p is initialized above if (AuthenticateUser(name, password, out roles) == true) {
foreach (string r in roles) { PrincipalPermission ppTemp = new PrincipalPermission(name, r);
p.Union(ppTemp); } } p.Demand();

- B. // PrincipalPermission p is initialized above if (AuthenticateUser(name, password, out roles) == true) {
 foreach (string r in roles) { PrincipalPermission ppTemp = new PrincipalPermission(name, r);
 p.Union(ppTemp); } } p.IsUnrestricted();
- C. if (AuthenticateUser(name, password, out roles) == true) { foreach (string r in roles) {
 Thread.CurrentPrincipal.IsInRole(r); } }Users will have access to different functionality depending on
- D. if (AuthenticateUser(name, password, out roles) == true) { Thread.CurrentPrincipal = new
 GenericPrincipal(new GenericIdentity(name), roles); }

Answer: D

For complete [Exam 70-340 Training kits and Self-Paced Study Material](http://www.Examsexpert.com/70-340.html)

Visit:

<http://www.Examsexpert.com/70-340.html>



www.Examsexpert.com



For Latest 70-340 Exam Questions and study guides- visit- <http://www.Examsexpert.com/70-340.html>