Module12

Building a Resilient ASP.NET MVC Web Application

Module Overview

- Developing Secure Sites
- State Management

Lesson 1: Developing Secure Sites

- Cross-Site Scripting
- Other Attack Techniques
- Disabling Attack Protection
- Secure Sockets Layer

Cross-Site Scripting

- Cross-site scripting involves:
  - Inserting malicious code in the session of a user
  - Posting information to other websites, without the knowledge of the concerned users
- You can prevent cross-site scripting by:
  - Using the **@Ajax.JavaScriptStringEncode** function

```
<div class="messages">@Ajax.JavaScriptStringEncode(ViewBag.Msg)</div>
```

  - Importing and using the AntiXSS library

```
@using Microsoft.Security.Application
<div class="messages">@Encoder.JavaScriptEncode(ViewBag.Msg)<div>
```

Other Attack Techniques

Cross-Site Request Forgery

To prevent this attack, you can:

- Use the **@Html.AntiForgeryToken()** function

SQL Injection Attack

To prevent this attack, you can:

- Validate user input
- Avoid using string concatenations to create dynamic SQL
- Use parameterized commands with dynamic SQL
- Store all sensitive and confidential information in encrypted formats
- Ensure that the application does not use or access the database with administrator privileges

Disabling Attack Protection

To protect your content from attacks, you can consider the following:

- Request validation helps determine potentially dangerous content
- Request validation can impede the performance of an application
- You can choose to disable request validation at different levels such as, in the Web.config file, on a webpage, or in a specific HTML element

Secure Sockets Layer

SSL:

- Encrypts content by using the public key infrastructure (PKI) keys
- Protects the content that is transmitted between the server and client
- Prevents unauthorized access of content during transmission
- Involves using the **RequireHttps** attribute to redirect users to the SSL link

Lesson 2: State Management

- Why Store State Information?
- State Storage Options
- Configuring State Storage
- Scaling State Storage Mechanisms
- Demonstration: How to Store and Retrieve State Information

Why Store State Information?

Session management:

- Enables web applications to store data for multiple HTTP requests
- Involves client-side session management techniques such as:
  - View state
  - Control state
  - Hidden fields
  - Cookies
  - Query strings
- Involves server-side session management techniques such as:
  - Application state
  - Session state
  - Profile properties
  - Database support

State Storage Options

- You can use the **TempData** object to store information relevant to requests
- Some commonly used state storage options include:
  - The InProc mode
  - The StateServer mode
  - The SQLServer mode
  - The Custom mode
  - The Off mode

Configuring State Storage

The StateServer Mode:

1. Run the ASP.NET state service on the server used for storing session information
2. Configure the ASP.NET application for using the StateServer mode
3. Configure StateServer support in your application

The SQLServer Mode:

1. Install the ASP.NET session state database on the SQL Server by using the aspnet_regsql.exe
2. Configure the ASP.NET application for using the SQLServer mode
3. Create the session state database on the SQL Server

Scaling State Storage Mechanisms

Partitioning the session state:

- Enables multiple state servers to handle state information
- Enables multiple SQL Server databases to handle state information
- Involves configuring the session management engine
- Involves implementing the **IPartitionResolver** interface

Demonstration: How to Store and Retrieve State Information

In this demonstration, you will see how to:

1. Store and retrieve values in a session state
2. Store a user preference for the background color by using the session state
3. Apply the background color preference to all pages in the web application

Lab: Building a Resilient ASP.NET MVC 4 Web Application

- Exercise 1: Creating Favorites Controller Actions
- Exercise 2: Implementing Favorites in Views

Estimated Time: 45 minutes

Lab Scenario

The senior developer has asked you to implement the following functionality in your Photo Sharing web application.

- Any visitor of the application, including anonymous users, should be able to mark a photograph as a favorite.
- If a user has marked a favorite, a link should be available to display the favorite photo.
- Favorite photos should be displayed in the slideshow view.

Lab Review

- In this lab, you stored the list of favorite photos in the session state. While testing, your manager notices that authenticated users lose their favorite photos list whenever they close their browser. Where would you store a list of favorites for each authenticated user so that the list is preserved whenever a user logs on to the web application?
- How would you create a view of favorite photos with the card-style presentation users see on the All Photos page?

Module Review and Takeaways

- Review Question(s)
- Real-world Issues and Scenarios