MS_logo_KMICROSOFT SDL - DEVELOPER STARTER KIT:

CROSS-SITE SCRIPTING (LEVEL 200)

Version 1.0

The following questions accompany the materials for the Microsoft SDL - Developer Starter Kit Cross-Site Scripting (Level 200) presentation.

For the latest information, please see [http://www.microsoft.com/sdl](http://go.microsoft.com/?linkid=9672761).

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in, or introduced into a retrieval system, or transmitted in any form, by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

© 2009 Microsoft Corporation. All rights reserved.

Microsoft, SQL Server, Visio, Visual Basic, Visual C#, Visual C++, Visual Studio, Windows, Windows Server, and Windows Vista are trademarks of the Microsoft group of companies.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Contents

[1.0 Microsoft SDL - Developer Starter Kit Content Comprehension Questions 2](#_Toc233605774)

[1.1 Introduction 2](#_Toc233605775)

[2.0 Cross-Site Scripting (Level 200) Questions 2](#_Toc233605776)

# 1.0 Microsoft SDL - Developer Starter Kit Content Comprehension Questions

## 1.1 Introduction

“The Microsoft Security Development Lifecycle (SDL) is an industry-leading software security assurance process. A Microsoft-wide initiative and a mandatory policy since 2004, the SDL has played a critical role in embedding security and privacy in Microsoft software and culture. Combining a holistic and practical approach, the SDL introduces security and privacy early and throughout all phases of the development process. It has led Microsoft to measurable and widely-recognized security improvements in flagship products, such as Windows Vista, Windows Server (2003 and 2008) and SQL Server. Microsoft is publishing the detailed SDL process guidance as part of its commitment to enable a more secure and trustworthy computing ecosystem.” -- [The Microsoft SDL 3.2 Whitepaper](http://go.microsoft.com/?linkid=9672762)

To help promote the adoption and awareness of the Microsoft SDL, Microsoft has developed content and demonstrations specifically for external developer audiences. The remainder of this document provides individuals who will present this content internally within their respective organizations with questions that may be used to ascertain comprehension of the subject matter addressed within the Microsoft SDL Training Module: Cross-Site Scripting (Level 200) presentation. These questions have been designed to enable the presenter to ascertain the extent at which the participating personnel with application development responsibilities have comprehended the subject matter addressed in the Cross-Site Scripting (Level 200) training module, as well as enabling the presenter to assess participants’ ability to apply the subject matter addressed to practical secure and trustworthy application development scenarios.

# 2.0 Cross-Site Scripting (Level 200) Questions

**Question #1:** Does the following code contain a cross-site scripting vulnerability?

protected void Page\_Load(object sender, EventArgs e)

{

String UserName = HttpUtility.HtmlEncode(Request.QueryString[“UName”]);

Response.Write(String.Format(“Your username is: {0}”,

HttpUtility.HtmlDecode(UserName)));

}

1. Yes, the un-trusted data read from the query string parameter UName is properly encoded; however, it is improperly decoded prior to embedding within Web responses.
2. Yes, no input validation is performed on the UName query string parameter.
3. No, the un-trusted data read from the query string parameter UName is properly encoded.
4. No, the call to String.Format removes any threat of a cross-site scripting attack.

**Answer:** The correct answer is “**A**”. The above code illustrates a very common implementation error where developers properly encode un-trusted data, but mistakenly decode the same data prior to embedding it in Web responses. Decoding encoded data has the effect of returning the un-trusted data to its original potentially executable form. Unlike cryptography where calls to encrypt data are eventually followed with calls to decrypt data, protecting applications and users from cross-site scripting attacks only requires that un-trusted data be encoded when included in Web responses.

**Question #2:** True or false: Technologies that protect data in transit, such as IPsec, SSL and SSH, can be used to remediate cross-site scripting vulnerabilities in Web-based applications.

1. True.
2. False.

**Answer:** The correct answer is “**B**” (false). Recall that cross-site scripting is an application-level attack. Any data protection that occurs below this level, such as at the network-level, does not affect a malicious user’s ability to exploit a cross-site scripting vulnerability.

**Question #3:** A verification team found an application behavior whereby the data entered by a user is stored in a database and then echoed back to other users without the use of encoding for Web responses. This potential vulnerability is an example of what type of cross-site scripting vulnerability?

1. A type 1, non-persistent cross-site scripting vulnerability.
2. A type 2, persistent cross-site scripting vulnerability.
3. This is not a cross-site scripting vulnerability.

**Answer:** The answer is “**B**”. Whenever un-trusted data is stored in a persistent data store, such as a database or file, and then embedded as part of a Web response later, a potential type2, persistent cross-site scripting vulnerability exists. If that un-trusted data contains malicious code or script, then any user visiting the vulnerable Web-application may be compromised. Answer “A” is incorrect since type 1, non-persistent cross-site scripting vulnerabilities do not store un-trusted data in order to execute an attack.

**Question #4:** In response to the cross-site scripting vulnerability discovered in Question #5, the development team proposed the remediation steps shown below. Which of these steps is appropriate to reduce the risk of cross-site scripting attacks?

1. Require SSL encapsulation of any un-trusted data.
2. Encode Web responses that may contain un-trusted data.
3. Require users to be authenticated at the network-level prior to accessing the Web-based application.
4. Require users to be authenticated at the application-level prior to accessing the affected Web-based application.

**Answer:** The correct answer is “**B**”. Data encoding renders executable code or script that may be embedded in un-trusted data into non-executable forms, and therefore answer “**B**” is correct. Answer “A” refers to the use of SSL to encrypt data to and from the Web application. This answer is incorrect because SSL occurs below the application layer and does not provide protection against cross-site scripting attacks. Similarly, the remediation step cited in answer “C” is also incorrect because it occurs below the application layer and will not provide any additional protection. Finally, authenticating users at the application layer, as suggested by answer d, will also be insufficient in protecting against cross-site scripting attacks. Authentication validates the identities of application users, but does not provide any assurance in terms of the type, range, format and length of data that is sent to and received from an application.

**Question #5:** The following ASP.NET code displays the ticket sales summary information from a database to Web users. The ticket sales database is populated with information imported from an outside entity. A developer working on this code has been advised by a security team that encoding is required in order to reduce the risk from cross-site scripting attacks. The developer is unsure as to what data should be encoded and asks you for help. Which of the following should be encoded?

protected void Page\_Load(object sender, EventArgs e)

{

String source = "Provider=ProviderInfo;

datasource=\\\\outsideserver\\exampleshare\\tickets.mdb";

String command="SELECT TicketNo FROM TicketsTable";

// Create OLE database command object

OleDbCommand myOleDbCommand = new OleDbCommand();

OleDbConnection myOleDbConnection=new OleDbConnection(source);

myOleDbConnection.Open();

myOleDbCommand.Connection=myOleDbConnection;

myOleDbCommand.CommandText=command;

// Execute the command object

OleDbDataReader mReader = myOleDbCommand.ExecuteReader();

// Read the data returned line by line

Response.Write("<h1>Ticket Sales Summary:</h1>");

while (mReader.Read())

{

Response.Write("<b>" + mReader.GetString(0) + "</b>");

}

// Cleanup

mReader.Close();

myOleDbConnection.Close();

}

1. There is no potential for a cross-site scripting attack to be conducted against this code so no encoding is required.
2. The string object name source should be encoded.
3. The String object named command should be encoded.
4. The String object returned from mReader.GetString(0) should be encoded.

**Answer:** The correct answer is “**D**”. Encoding should be performed on any un-trusted data that is embedded in a Web response just prior to writing the Web response. The string objects, source and command, contain information with which to retrieve the ticket information, and do not get embedded in Web responses. Finally, the tickets database is created by an outside entity and may potentially contain malicious script data. Since the data contained within the tickets database is embedded as part of a Web response, the threat of a cross-site scripting attack exists; therefore, answer “A” is incorrect.