MS_logo_KMICROSOFT SDL - DEVELOPER STARTER KIT:

SECURITY CODE REVIEW (LEVEL 200)

Version 1.0

The following questions accompany the materials for the Microsoft SDL - Developer Starter Kit Security Code Review (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](#_Toc233627207)

[1.1 Introduction 2](#_Toc233627208)

[2.0 Security Code Review (Level 200) Questions 2](#_Toc233627209)

# 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: Security Code Review (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 Security Code Review (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 Security Code Review (Level 200) Questions

**Question #1:** Why is there a buffer overflow vulnerability in the code shown below?

void CopyBuffer(char \* src)

{

char Buffer[32];

strncpy(Buffer,sizeof(Buffer),src);

Buffer[sizeof(Buffer)]=‘\0’;

}

1. There is no buffer overflow vulnerability shown in the code above.
2. The last element of Buffer is calculated incorrectly.
3. There is a mismatch between the input parameter (char \*) and the character array Buffer (char[]).

**Answer:** The correct answer is “**B**”. The call to strncpy restricts the size of the copy to at most the size of Buffer, which is 32 bytes; however, the last instruction in this function is where the vulnerability can be found. The last statement is meant to terminate the Buffer variable by setting the last element in the array to a null-terminating character. The index of the last element, however, is calculated incorrectly. In C and most other programming languages, arrays are zero-based. This means that the Buffer array in the code example starts at index 0 and ends at 31. The call sizeof(Buffer) will return 32, so the code above is actually writing a null-termination character 1 byte beyond the bounds of the Buffer variable, resulting in a buffer overflow of exactly 1 byte.

**Question #2:** True or false: security code review can be used to replace code scanning tools and vice versa?

1. True.
2. False.

**Answer:** The correct answer is “**B**” (False). Security code review has the distinct disadvantage that it is a highly manual process. This means that for large application source code bases security code review may not be feasible or practical. Alternatively, code analysis tools can scale across large application code bases; however, they suffer from false-positives and false negatives. As such, security code review should not be used to replace code analysis tools and vice versa.

**Question #3:** A web developer has asked you to review her code that is shown below. What type of vulnerability is present in this code?

<%@ Page Language="C#" ValidateRequest="false" %>

<html>

<script runat="server">

void btnSubmit\_Click(Object sender, EventArgs e)

{

Response.Write(nameString.Text);

}

</script>

<body>

<form id="form1" runat="server">

<asp:TextBox id="nameString" runat="server"/>

<asp:Button id="btnSubmit" runat="server"

OnClick="btnSubmit\_Click"

Text="Submit" />

</form>

</body>

</html>

1. SQL injection.
2. Web-based buffer overflow.
3. Cross-site scripting.
4. There is no vulnerability present in this code.

**Answer:** The correct answer is “**C**”. A cross-site scripting vulnerability is present in this code and can be exploited whenever the value of the text box with the id nameString contains executable script, such as “<script>…</script>” and the button with id btnSubmit is triggered. Additionally, the ValidateRequest ASP.NET feature is disabled, which prevents ASP.NET from automatically checking and protecting applications from dangerous inputs.

**Question #4:** The following code reads a 10 digit account number from a Web resource query string parameter and echoes the data read as part of a Web response. Does this code contain a cross-site scripting vulnerability?

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

{

String AccountNumber = Request.QueryString[“AccountNumber”];

if (!Regex.IsMatch(AccountNumber, @”^\d{10}$”))

{

return;

}

Response.Write(“Your account number is: “+ AccountNumber);

}

1. Yes, the AccountNumber data is not encoded prior to embedding it in a Web response.
2. Yes, a malicious user could embed malicious script in the AccountNumber query string parameter value, which will be embedded as Web response.
3. No, the input validation performed by the regular expression ensures that the AccountNumber consists of 10 digits only and properly validates for type, length, range and format.
4. No, protected methods safeguarded from cross-site scripting vulnerabilities and attacks.

**Answer:** The correct answer is “**C**”. The method shown above, while it does not encode the AccountNumber prior to embedding it in Web responses, performs proper input validation on AccountNumber for type, length, range and format. The regular expression used helps to ensure that the AccountNumber provided contains only digits and that AccountNumber is exactly 10 digits in length with no spaces, alpha characters or other non-digit data. In this case, the input validation performed is sufficient to help ensure that the above method is not susceptible to a cross-site scripting attack.

**Question #5:** A security assessment team has been assigned to review a rather large application code base that consists of approximately 5 million lines of code. The security assessment team wisely decides to review the code by looking for specific vulnerabilities in one pass, and then focusing on different vulnerabilities in subsequent passes. Also, in order to reduce the amount of human error into final results, different reviewers will be employed for each pass. The security code review strategy that the security assessment team is using is called:

1. A multi-pass approach.
2. An expert-level review.
3. A progressive review.
4. A vulnerability-centric review.

**Answer:** The correct answer is “**A**”. A multi-pass approach means that security code reviews are performed by conducting several passes. In each pass, a specific type of vulnerability is sought after and different reviewers are used to reduce the amount of human error introduced into the review process.