MS_logo_KMICROSOFT SDL - DEVELOPER STARTER KIT:

CODE ANALYSIS (LEVEL 200)

Version 1.0

The following questions accompany the materials for the Microsoft SDL - Developer Starter Kit Code Analysis (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](#_Toc233596456)

[1.1 Introduction 2](#_Toc233596457)

[2.0 Code Analysis (Level 200) Questions 2](#_Toc233596458)

# 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: Code Analysis (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 Code Analysis (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 Code Analysis (Level 200) Questions

**Question #1:** Microsoft PREFast is an example of what type of code analysis tool?

1. A static source code analysis tool.
2. A binary analysis tool.
3. A fuzz testing tool.

**Answer:** The correct answer is “**A**”. Microsoft PREFast analyzes the source code implementation of C and C++ applications and is therefore an example of a static source code analysis tool.

**Question #2:** True or false: Code analysis tools can be used as a replacement for security implementation and verification best practices

1. True.
2. False.

**Answer:** The correct answer is “**B**” (False). While code analysis tools can be used to analyze large application source code bases, code analysis tools can not ensure that all implementation vulnerabilities will be identified. In fact, no one security assessment tool or technique can provide such assurance. Therefore, application development teams should use code analysis tools in conjunction with other sound security implementation and verification best practices and techniques.

**Question #3:** An application development team that wishes to analyze the source code implementation of an application written in C++ can use which of the following code analysis tools?

1. Microsoft PREFast.
2. Microsoft Visual Studio code analysis tool (/analyze).
3. Answer a or b.
4. Microsoft FxCop.

**Answer:** The correct answer is “**C**”. Applications written in C++ can be analyzed with either Microsoft PREFast or the Microsoft Visual Studio code analysis feature (/analyze), making “**C**” the correct answer. Microsoft FxCop is incorrect because it is a binary analysis tool that analyzes compiled .NET Framework assemblies.

**Question #4:** A code analysis tool reported that a buffer overflow vulnerability exists in an application’s source code implementation. However, upon closer inspection of the reported vulnerability, the vulnerability reported was in fact not a vulnerability. This is an example of what type of automated code analysis outcome?

1. A false-positive.
2. A false-negative.
3. A language-centric finding.
4. Answer a and c.

**Answer:** The correct answer is “**A**”. When a code analysis tools reports a vulnerability that is in fact not a vulnerability, this is what is known as a false-positive. A false-negative is a potential code vulnerability that is not reported and thus answer b is incorrect. Answer “C” refers to vulnerabilities that are unique to certain programming languages and is also an incorrect response for this question.

**Question #5:** A code analysis tool was used to analyze the implementation of an application and reported that no vulnerabilities were present. The application was released to customers, and a month later a SQL injection vulnerability was publicly reported and subsequently exploited. This is an example of what type of automated code analysis outcome?

1. A false-positive.
2. A false-negative.
3. A language-centric finding.
4. Answer a and c.

**Answer:** The correct answer is “**B**”. Whenever a code analysis tool fails to report the presence of a vulnerability that exists in an application’s implementation, this is known as a false-negative. Situations like the one described in this question are not as uncommon as people may think and provides further justification as to why code analysis tools cannot be used as replacements for other security verification techniques, such as security code review and fuzz testing.