MS_logo_KMICROSOFT SDL - DEVELOPER STARTER KIT:

SECURE VERIFICATION PRINCIPLES (LEVEL 100)

Version 1.0

The following questions accompany the materials for the Microsoft SDL - Developer Starter Kit Secure Verification Principles (Level 100) 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](#_Toc233624449)

[1.1 Introduction 2](#_Toc233624450)

[2.0 Secure Verification Principles (Level 100) Questions 2](#_Toc233624451)

# 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: Secure Verification Principles (Level 100) 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 Secure Verification Principles (Level 100) 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 Secure Verification Principles (Level 100) Questions

**Question #1:** True or false: Functional testing already includes security testing

1. True.
2. False.

**Answer:** The correct answer is “**B**” (False). Functional testers evaluate to what extent an implemented application adheres to its original design. Security testers also focus on to what extent an implemented application adheres to its original design; however, security testers also focus on identifying the presence of the unintended functionality of an application that a malicious user may try to exploit.

**Question #2:** An application development team is worried about information disclosure threats and has come to you for advice as to which security verification approaches they should pursue. The application development team has resources to conduct only two of the three security verification approaches listed below. Which of the following approaches would you not recommend?

1. Fuzz testing.
2. Penetration testing.
3. Security code review.

**Answer:** The correct answer is “**A**”. Fuzz testing is an excellent security verification approach and it has helped Microsoft greatly improve the safety of its products and services; however, in order for fuzz testing to be effective an application needs to crash or throw an exception in some unexpected or unhandled way. Information disclosure threats typically do not cause an application to crash and therefore fuzz testing would not be effective in this case.

**Question #3:** A security verification tool that analyzes the compiled files of an application is an example of what type of security automation tool?

1. Static source code analysis.
2. Binary analysis.
3. Run-time analysis.

**Answer:** The correct answer is “**B**”. Binary analysis tools examine the compiled form of an application for vulnerabilities.

**Question #4:** An application development team is in the initial planning stages of the software development lifecycle and is planning to conduct a penetration test at some point. At what stage within the software development lifecycle would you recommend the application development team conduct the penetration test?

1. When an application is being initially designed.
2. When an application first successfully compiles.
3. When an application is nearing its final state.

**Answer:** The correct answer is “**C**”. The results from a penetration test are relevant to the application at the time that the test was conducted. If an application changes or is re-engineered in any way, then the results from the previously conducted penetration test will often no longer apply. Therefore, the application development team should perform the penetration test when the application is nearing its final state.

**Question #5:** An application development team ahead of time agrees upon how vulnerabilities will be addressed by severity levels. Vulnerabilities and threat conditions are grouped by severity and addressed according to the assigned severity level. This is an example of what type of strategic activity?

1. A security test plan.
2. A security bug bar.
3. A penetration test plan.
4. An application dependency assignment.

**Answer:** The correct answer is “**B**”. A security bug bar defines the severity of a security vulnerability or threat condition and the criteria required to be considered in that severity category. Security vulnerabilities and threats with higher severity ratings are addressed with higher priority than those with lower severity ratings.