MS_logo_KMICROSOFT SDL - DEVELOPER STARTER KIT:

SECURE DESIGN PRINCIPLES (LEVEL 100)

Version 1.0

The following questions accompany the materials for the Microsoft SDL - Developer Starter Kit Secure Design 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](#_Toc233613840)

[1.1 Introduction 2](#_Toc233613841)

[2.0 Secure Design Principles (Level 100) Questions 2](#_Toc233613842)

# 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 Design 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 Design 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 Design Principles (Level 100) Questions

**Question #1:** Attack surface is any part of an application that is accessible by:

1. A human.
2. Another application.
3. Both humans and other applications.

**Answer:** The correct answer is “**C**”. The attack surface of an application is the portion of the application that is exposed to external entities, such as humans and other applications. While both “A” and “B” captures aspects of attack surface, the more encompassing answer is “**C**”.

**Question #2:** The goal of attack surface reduction is to:

1. Minimize the number of exposed attack surface points, which with a malicious user can discover and attempt to exploit.
2. Minimize the number of vulnerabilities present in an application.
3. Maximize the number of security code scanning tools used to analyze an application’s source code implementation.

**Answer:** The correct answer is “**A**”. The goal of attack surface reduction is to reduce the number of possible entry points that a malicious user can access and attempt to exploit. Reducing the number of vulnerabilities in an application, as indicated in answer ”B”, is the goal of secure coding and not attack surface reduction. Finally, similar to answer ”B”, the use of security code scanning tools will help developers identify vulnerabilities in code and reduce the overall number of vulnerabilities present in an application, but will not reduce the attack surface.

**Question #3:** True or false: The goals of privacy and security are the same.

1. True.
2. False.

**Answer:** The answer is “**B**” (False). Privacy focuses on the control and choices a user has regarding the use, collection and distribution of their personal information. Security is applied to an application to help protect assets, including personal information, from threats. Thus the goals of each are not the same.

**Question #4:** True or false: Threats and vulnerabilities refer to the same thing.

1. True.
2. False.

**Answer:** The correct answer is “**B**” (False). A threat is what an adversary might try to do to compromise a protected resource on a system. A vulnerability, however, is the specific way that a threat is exploitable based on an unmitigated attack path, and therefore the answer to this question is false.

**Question #5:** Applying the principle of Defense in Depth means to:

1. Design an application with a single layer of defense at the perimeter of an application, such as network sockets and user interface points.
2. Design an application with a single layer of defense at the point where data from an un-trusted source is consumed by an application.
3. Design an application with multiple layers of defense so that if one fails another layer can still provide protection.

**Answer:** The correct answer is “**C**”. Designing an application with defense in depth means to design an application in such a way that if one defense layer fails, then there are additional layers of defense that can provide protection to that application. Both answers “A” and “B” indicate a single layer of defense, which is contrary to the Defense in Depth principle.

**Question #6:** The goal of the principle of Least Privilege is to:

1. Reduce the number of vulnerabilities in application source code.
2. Limit the potential damage that may be inflicted should a malicious user compromise an application.
3. Help application designers select the best authentication technology to use.

**Answer:** The correct answer is “**B**”. With the Least Privilege principle, it is assumed that an application will be compromised at some point by a malicious user. However, by deploying applications with the minimum access to function correctly, the potential damage that can be inflicted due to a compromise is contained and isolated. The Least Privilege design principle will not help to reduce the number of vulnerabilities in application source code, nor will it help to select application designers choose one authentication technology over another. Therefore, answers “A” and “C” are incorrect.

**Question #7:** The goal of the principle of Secure Defaults is to:

1. Better ensure that customers get safer experience with your application after extensive configuration.
2. Ensure that applications are pre-configured with default security passwords.
3. Better ensure that customers get safer experience with your application out of the box and without extensive configuration.

**Answer:** The correct answer is “**C**”. With the Secure Defaults principles, applications are designed so that by default they are configured with more secure settings. This alleviates the need for requiring users to commit extensive manual efforts to deploy applications more securely; therefore, answer “A” is incorrect. Answer “B” is incorrect, and is in fact contrary to the principle of Secure Defaults. By deploying applications with default security passwords, organizations that use those applications without changing those default passwords are left exposed to attack by malicious users who are aware of those default passwords.

**Question #8:** A security team has threat modeled an application design and has found various threats that have not been addressed. Of the threats identified, the most serious threat encountered describes the ability of a malicious user to elevate their privileges via a feature that is running in a SYSTEM security context. This means that if this feature were to be compromised, a malicious user could potentially gain unfettered access to the rest of the host system and internal network. One option to address this threat is to simply remove the affected feature; however, the application design team has indicated that this is not possible. Which of the following secure design principles could be applied to reduce the risk from this potential threat if removing the affected feature is not possible?

1. Re-design the application so that it runs in a security context with the minimum required access it needs to function correctly.
2. Re-design the application so that it runs in a security context of a domain administrator.
3. Re-design the application so that all remote interaction with the application is done over secure socket layers (SSL).

**Answer:** The correct answer is “**A**”. Running the application in a security context with the minimum access required for the application to run correctly is an example of the Least Privilege design principle and will limit the access gained by a malicious user if the application feature is compromised. Running an application as a domain administrator account has essentially the same effect as running the application in a local host SYSTEM context and will not reduce the overall risk. Finally, forcing all remote interaction with the application over SSL does not reduce the overall risk. SSL affects how data is delivered to the application, but not how it is processed by the application and therefore will not reduce the risk from the threat described above.

**Question #9:** An application saves sensitive user data to an XML file on a host system. The file cannot be read by anonymous users, but can be read by any user that is authenticated on the host system. According to the application documentation, if an organization wants to add additional protection for this file they can change the file system access control lists (ACLs) so that only the security context that is used to start the application and local administrators can access this file. The application documentation further provides the manual configuration steps (approximately 5-7 steps) necessary to accomplish this. Which secure design principle does this scenario violate?

1. Threat modeling.
2. Least Privilege.
3. Defense in Depth.
4. Secure Defaults.

**Answer:** The correct answer is “**D**”. Applications that built with the Secure Defaults principle are installed with the most secure settings already configured. In the scenario above, the end user needs to make additional configuration settings after installation to ensure that the application is running the most secure state as possible, which is contrary to the Secure Defaults design principle.