SECURE PROGRAMMING FOUNDATION

Sample Exam Questions

SECO INSTITUTE

Introduction

This document contains 5 questions (and answers) that help you familiarise yourself with the structure and topic areas of the SECO-Institute's Secure Programming Foundation certification exam.

To download our Complete Sample Exam, create a free study account at https://members.seco-institute.org

We recommend you to take the Complete Sample Exam before registering for the certification exam.

The results of the Sample Exam do not count towards your examination score.

Certification exam

You can book your exam with an accredited training partner or directly with the SECO-Institute. Attending a course is not a prerequisite for taking a certification exam.

To book an exam with the SECO-Institute, go to: https://www.seco-institute.org/how-to-book-your-exam-schedule-an-exam/

By passing the certification exam and earning a SECO-Secure Programming Foundation Certificate, you demonstrate that you are aware of the most common causes of software vulnerabilities. You understand how attackers exploit software vulnerabilities, and you know how to prevent software flaws that enable cyberattacks.

Exam format

Computer-based with remote proctoring

- 40 multiple-choice questions
- Time allowed: 60 minutes
- Closed-book exam
- Pass mark: 60%

Questions

Question 1

What is the best answer to the question: Why do we have insecure software?

A. Consumers cannot objectively assess the quality and security of available software.
B. Software companies can sell more software if they offer more features, lower prices and faster delivery than their competition.
C. We are bad at estimating risks. Consequently, we are unable to estimate the odds that the software built for us is vulnerable and we are unable to predict the resulting damage if those vulnerabilities are exploited.
D. Consumers focus on price and features and software vendors are not liable for insecure products.

Question 2

To fortify trust boundaries and improve overall security, web applications should be divided into clear modular components. Yet this is difficult to arrange. Which answer best explains why modular design is difficult to implement?

A. In today's web applications, JavaScript and applets are called within the browser but these components communicate directly with the web server.
B. Most users score web applications only on performance. This contradicts modular design, in which the code amount is limited by building multi-purpose routines.
C. The clear communication line between web client and web server can be blurred by a so-called man in the middle (MITM) proxy.
D. Modern applications can contain millions of lines of code. The resulting code complexity impedes modular design.

Question 3

You will read 4 statements about GET requests. Which statement is closest to the truth?

A. GET requests are one of the three original HTTP Request types distinguished in the CGI standard: GET, POST and CONNECT.
B. The original semantics of the CGI standard require that a GET request modify the application state.
C. GET parameters are visible in URL, and therefore in the Browser address bar and in various logs.
D. The original semantics were forgotten and today both GET and POST requests are equally secure.

Question 4

Which command is an example of a parameterized query (not vulnerable to SQL injection)?

A. [C#] SqlCommand cmd = new SqlCommand("SELECT * FROM people WHERE LastName ='" +LastName.Text+"')", conn);
B. [PHP] $query = "SELECT * FROM people WHERE LastName =".$_POST['LastName'];
C. [Python] cmd = "SELECT * FROM people WHERE LastName ='%s'" % (LastName)
D. [JDBC] PreparedStatement statement = connection.prepareStatement("SELECT * FROM people WHERE LastName = ?"); statement.setString(1, LastName);

Question 5

To avoid SQL injection, what is the MAIN difficulty for a programmer to neutralize metacharacters?

A. Routines that are provided by the system may be buggy.
B. There are many metacharacters, so it is easy to miss a few.
C. It is not a good idea to write your own escaping routines.
D. OWASP's ESAPI library only links to a few database dialects and was not yet properly reviewed.

Answers

Question	Answer	Explanation
1	D	"Consumers focus on price and features and software vendors are not liable for insecure products" is the only answer which contains 2 cause-and-effect arguments.
2	A	If programmers are unaware that their browser component directly communicates with the web server, they may forget to address security in this component.
3	C	The (security) differences between GET and POST are that GET parameters are visible in URL (and therefore in the Browser address bar and in various logs), while a POST request modifies the application state.
4	D	[JDBC] PreparedStatement statement = connection.prepareStatement("SELECT * FROM people WHERE lastName = ?"); statement.setString(1, name); //lastName is a VARCHAR
5	B	Since there are many metacharacters, it is easy to miss a few. An important one that should not be forgotten is the escape character itself. Therefore, it is usually a bad idea to write your own escaping routines.