



Daffodil International University
Faculty of Science and Information Technology
Department of Information Technology & Management
Mid-Term Examination, Summer 2026
Course Code: ITM 322; Course Title: Software and Web Security
Sections & Teachers: A (AR)

Time: 1:30 Hours

Marks: 25

Answer ALL Questions

[The figures in the right margin indicate the full marks. All portions of each question must be answered sequentially.]

1.
 - a) State the concept of Information System Security (ISS). Differentiate between snooping and spoofing with examples. [Marks-3]
 - b) List the key steps in the Cyber Security Process to protect systems from potential threats. Briefly summarize the tools to attain confidentiality. [Marks-3]
 - c) Explain how a firewall works to protect a network from unauthorized access with its working procedure. [Marks-3] **CO-1 Level-4**
 - d) Describe the designing of a secure web application by applying the major phases of the Secure Software Development Life Cycle (SecSDLC). [Marks-3]
 - e) Analyze how a rootkit operates, including its common attack mechanisms, and recommend appropriate mitigation strategies to detect and prevent rootkit attacks. [Marks-3]
2.


```

<?php
$conn = mysqli_connect(
    "localhost", "root", "", "banking"
);

$account_no = $_GET['account_no'];

$sql = "SELECT transaction_id, account_no,
        amount, transaction_type
        FROM transactions
        WHERE account_no = '$account_no'";

$result = mysqli_query($conn, $sql);

while ($row = mysqli_fetch_assoc($result)) {
    echo $row['transaction_id'] . " - "
        . $row['amount'] . " - "
        . $row['transaction_type'] . "<br>";
}
?>

```

CO-2 Level-6

- a) Identify the vulnerable line(s) in the given scenario and explain the security risks. Write 2 SQL injection payloads that could be entered into the public account_no field in an authorized testing environment. Briefly explain the intended effect of each payload. [Marks-5]
- b) Rewrite the vulnerable PHP code using prepared statements and parameter binding to prevent SQL injection. Describe how your revised implementation prevents user input from altering the intended SQL query. [Marks-5]

ourdiu.com