Information Security (CS3002)

Sessional-II Exam

Total Time (Hrs): 1
Total Marks: 40
Total Questions: 4

Date: Nov 4th 2025

Course Instructors

Ahmad Ali Shah, Ammar Haider, Aqsa Khalid, Nosheen Manzoor, Zeeshan Ali Khan

SOLUTION

_____	_____	_____
Student Roll No	Section	Student Signature

Instructions:

1. Attempt all questions on the answer booklet. Do not write with a lead pencil.
2. If you think some information is missing then make an assumption and state it clearly.

CLO-1	Explain key concepts of information security such as design principles, cryptography, risk management.
CLO-2	Discuss legal, ethical, and professional issues in information security.
CLO-3	Analyze real world scenarios, model them using security measures, and apply various security and risk management tools for achieving information security and privacy.
CLO-4	Identify appropriate techniques to tackle and solve problems of real life in the discipline of information security.
CLO-5	Describe issues related to ethics in the field of information security.

CLO 1

Q1: Multiple choice questions **[6 marks]**

1. In a healthcare database, queries with fewer than three matching records are automatically rejected. This mechanism primarily prevents:
   a) SQL injection
   b) Data perturbation
   c) Inference of sensitive information
   d) Schema normalization

2. In Kerberos, a client will interact with the Ticket Granting Server (TGS) _____.
   a) once per logon
   b) once per service
   c) every time services are used
   d) None of the above

3. Attackers exploit integer overflow vulnerabilities by _____.
   a) taking advantage of poorly readable code
   b) feeding unexpected inputs to arithmetic operations
   c) injecting malicious instructions directly into the source code
   d) slowing down code execution speed

4. Which of the following mitigations is most effective against SQL injection?
   a) Escaping all user input manually
   b) Obfuscating the database schema
   c) Using dynamic SQL queries
   d) Using prepared statements

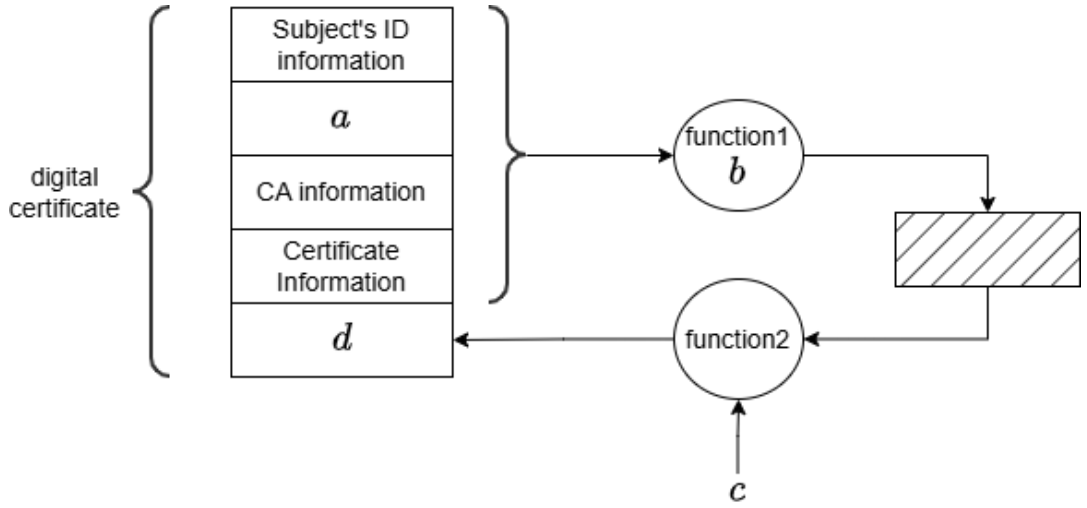
5. Hackers stole thousands of hashed passwords from a company database and noticed that two users, Peter and Sam, had identical hashes—indicating they used the same password. What is the best defense the company can implement to prevent attackers from discovering which users share the same passwords?
   a) Stronger Hashing
   b) Password encryption
   c) Password salting
   d) Mandatory long passwords

6. Which statement correctly distinguishes a virus from a worm?
   a) A virus self-replicates over networks without user action; a worm attaches to host files and needs user action.
   b) A virus is always remote-access software; a worm always encrypts files.
   c) A virus typically needs a host file and often requires user action to spread; a worm self-replicates across networks without user interaction.
   d) Virus and worm are just different names for ransomware.

CLO 3

Q2: Short answer questions **[4 + 6 + 3 + 2 + 2 marks]**

A. The diagram below shows the high-level process of creating a digital certificate. Complete it by identifying the missing fields (a) and (d), the function1 used at point (b), and the input (c) of function2.



(a) subject's public key (b) hash function (c) CA's private key (d) digital signature by CA

B. For each of the following scenarios, identify the most appropriate malware category and give a one-sentence justification for your choice.

i. A new virus somehow enters the system. Once active, it identifies and disables the antivirus process, blocks system updates, and prevents access to antivirus vendor websites.

ii. A freelancer installs a free "time tracking" app to monitor productivity. Unknown to them, it secretly logs keystrokes, screenshots, and browser activity. Weeks later, their email and banking accounts are compromised.

iii. A hacker gains admin access through a phishing link and installs a software that hides malicious files and processes. Even system logs and antivirus scans show no signs of infection. The attacker silently collects confidential data for weeks before detection.

i. Retro virus since it is blocking/interfering with security software
ii. Spywayre for information stealing behaviour
iii. Rootkit due to highly stealthy operations

C. Your learnt about two methods of database encryption: field-level encryption and indexed record-level encryption. Discuss how these two methods differ in terms of data granularity and query efficiency.

Field-level is very granular because each individual attribute within a table, such as name, salary, and CNIC is encrypted separately, offering very fine-grained control and ability to query individual attributes. But this approach significantly reduces query efficiency because encrypted values cannot be easily searched, sorted, or compared, making range queries (e.g., "salary > 70000") nearly impossible without decrypting individual fields.

Record-level encryption encrypts an entire row of data as a single unit, securing all its fields together. This limits selective access (every query will download full rows). Indexing the attributes makes searching quicker, allowing complex WHERE clauses on encrypted records.

D. In the context of Challenge-Response authentication using symmetric cryptography:

- What problem does this method address?
- What is the main drawback of this approach?

Addresses 'remote authentication' problem, where passwords can not be simply transmitted due to risk of attacker eavesdropping.

Drawback: server needs to store all secrets in cleartext. In case of database hack, all users' secrets will be disclosed in one go.

E. A high-security facility uses facial recognition for access control. The security team is worried about unauthorized users gaining entry, so they decide to use a similarity score threshold of 99%. What could be a negative consequence of using such a high threshold? Discuss with reasons.

Using such a high threshold will have benefit of minimizing false positives (no unauthorized users gaining access). But negative consequence will be high false negatives – authorized users denied access. Reason: biometric template matching is always fuzzy – no two templates of the same user will match 100%.

CLO 4

Q3: Practical problems **[3 + 4 + 3 marks]**

A. A user logs in to shopping.com and remains authenticated. Later, he receives an HTML email containing the following image tag:

```
<img src="https://shopping.com/deleteAccount?confirm=yes">
```

When the user's email client loads the image, his account is deleted. Explain what is this vulnerability and propose a suitable defense.

This is a CSRF attack exploiting an authenticated session to perform an unintended action i.e. account deletion. The browser automatically sends cookies with the GET request.

Defense: Checking Origin/referrer headers, or implementing STP tokens.

(Note: generally state-changing requests should be sent as POST, not GET as given in this question).

B. For each of the scenarios below, name the SQL injection category that best matches the attack.

i. An attacker appends `; DROP TABLE ...` to an input so the server executes multiple SQL statements.
ii. An attacker sends `OR 1=1` in an input to bypass authentication.
iii. An attacker submits Boolean test conditions and observes page behavior (true/false) to establish if injection is working or not.
iv. An attacker injects SQL into a web request and sees the resulting query output in the web response.

i. Piggybacked query
ii. Tautology
iii. Blind injection
iv. In-band attack

C. The given server-side code is used on the welcome page of a web application. An attacker sends the following link to a victim:

```
if ($_GET['user']) {
  echo "Welcome ".$_GET['user'];
}
```

http://victim.com/profile.php?user=<script>alert('hack')</script>

Identify the **specific** vulnerability that attacker is hoping to exploit. Also explain how that vulnerability can be mitigated by the developer.

It is Reflected XSS vulnerability.

Developer should validate the 'user' parameter value, and also escape the data before inserting into output webpage.

CLO 4

Q4: Practical problems **[2 + 2 + 3 marks]**

The following code runs on a 32-bit computer, which means that the machine handles data in units of 32 bits (4 bytes). The figure on the right depicts the stack at point X before the function `gatekeeper()` returns. The stack grows downwards towards lower addresses.

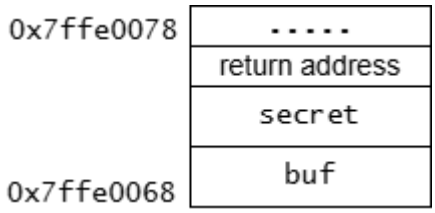
```
void fill_secret(char *);
void unlock(void);
void backdoor(void);

void gatekeeper() {
    char secret[6];
    char buf[6];

    fill_secret(secret);
    gets(buf);

    if (strncmp(buf, secret, 6) == 0)
        unlock();

X   return 0;
}
```



As a hacker, your task is to gain access to the machine. If you try to brute force the password, you will be locked out. `gatekeeper()` function compares a password you provide with the system's password. `fill_secret()` function places the system's password into the `secret` buffer so that it can be compared with the user-provided password stored in `buf`.

A. Explain why the use of the `gets()` function introduces a security vulnerability in the program.

gets() introduces a security vulnerability because it does not have a limit on the number of characters read. This can cause a buffer overflow, allowing a user to enter a malicious string that exceeds our buffer capacity.

B. You wish to unlock the machine without the correct password. Provide a hexadecimal representation of an attack string that causes the `strncmp()` call to return 0 so that `unlock()` is then called. `gatekeeper()` should return normally, as to avoid raising any suspicion.

Any input that makes the buffers 'secret' and 'buf' same.

For example, `0xffffffffffabffffffffffab`

C. The function `backdoor()` is located at address 0x00000351. Construct a string that can be inputted to this program that will cause the function `gatekeeper()` to unconditionally transfer control to the function `backdoor()`. Provide a hexadecimal representation of your attack string, also explaining why it will result in transfer of control.
Data can be stored in memory using either little-endian or big-endian order.

Attacker needs to fill up buf (6 bytes), fill up secret (6 bytes) with same value, and replace return address with backdoor's address (4 bytes).

For example: `0x 123456781234 123456781234 00000351`

Last four bytes can also be in little-endian order: `51030000`