16.5
7
14

National University of Computer and Emerging Sciences, Lahore Campus

Course Name:	Information Security	Course Code:	CS3002
Program:	BS (CS, SE, DS)	Semester:	Fall 2023
Duration	60 minutes	Total Marks:	35
Date:	14-11-23	Weight	12.5
Exam Type:	Midterm-II	Pages	6

Student : Name: Hashim Bilal Roll No. 20L-1219 Section: BCS-7B

Instruction: If you think some information is missing then make an assumption and write it clearly.

Question 1: [CLO:1] [5 marks]

1. ______ malware type does not modify the total size of infected file.
   - A. prepending
   - B. appending
   - C. overwriting
   - [✓ D. cavity]

2. In Kerberos protocol, the Ticket-Granting Server (TGS) issues ______ to clients.
   - A. ticket-granting tickets
   - B. verification tickets
   - [✓ C. service tickets]
   - D. correct-user tickets

3. Which one of the following characters is most important to restrict when performing input validation to protect against XSS attacks?
   - A. &
   - B. !
   - [✓ C. <]
   - D. $

5

4. What is the best design for input validation?
   - A. Detecting attacks and rejecting them
   - [✓ B. Setting a policy for good input and rejecting everything else]
   - C. Setting a policy for bad input and logging them
   - D. None of the above

5. Random Sample Query is a ______ perturbation technique to protect from ______ attack.
   - A. Data, In-band
   - B. Output, In-band
   - C. Data, Inferential
   - [✓ D. Output, Inferential]

0088

7 [CLO: 3] [4+2+2+2 ma

Question 2: Short Questions

a. Write SQL statements (DB access control) for each of the following tasks. (4 marks)

i. Allow a user 'peter' to create new rows or delete some rows in 'Repository' table. He should not be able to see table data or modify anything within the table rows.

ii. User 'sam' has full access for all tables in databases. Take away his writing privileges, but keep the reading access.

(i). GRANT ~~CREA~~ INSERT, DELETE on any table on Pe

3 (ii). REVOKE INSERT, UPDATE, DELETE on any table on Sam

b. "In salted password storage, the salt should not be leaked to attackers in any case, otherwise user's password will be compromised." Discuss whether you agree or disagree with the statement.

Yes, I agree, the salt should not be leaked to attackers in any case, since salt is the key of encryption for the password, which is randomly generated. and can decrypt user password.

c. Discuss how checking the Origin header can help in mitigating CSRF attacks.

2 The origin header if present in requests, its value is matched with the value of target origin header on server side, if it matches, it means request is not an attack, otherwise, if ~~ma~~ not matched, it means request is an attack. Hence, CSRF attacks can be mitigated by such checking of Origin header.

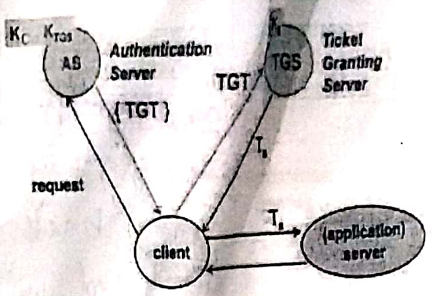
d. What kind of obfuscation techniques is used by retro viruses to avoid detection and analysis?

The obfuscation techniques are: 2

1. Using anti-debugging, anti-VM techniques.
2. Blocking Anti-virus from updating.
3. Hindering operations of anti-virus
4. Hacking and changing anti-virus database of viruses.

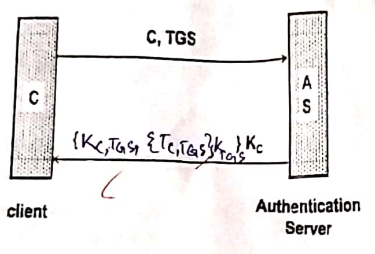
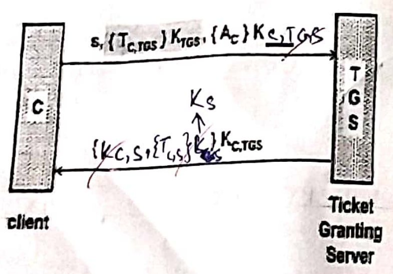
[CLO: 1] [6 marks]

tion 3:
e following figure depicts the Kerberos Authentication proces with keys involved.

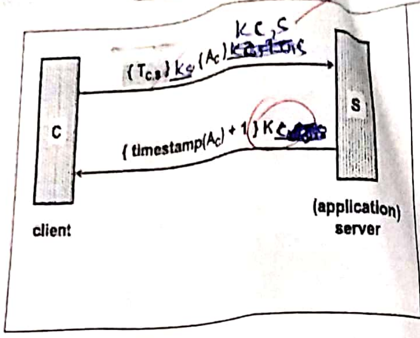


5.5

If a client wants to access a specific service from application server then how the user will be authenticated? Partial information is provided in the following figures. Complete the missing information and explain each process in the right column.

 C, TGS C A S $\{K_{C,TGS}, \{T_{C,TGS}\}K_{TGS}\}K_C$ client Authentication Server	A.S returns TGT to Client encrypted with key of TGS, and session key of user for TGS, both these are encrypted with public key of client
 $s, \{T_{C,TGS}\}K_{TGS}, \{A_C\}K_{C,TGS}$ C T G S K_S $\{K_{C,S}, \{T_{C,S}\}K_{S}\}K_{C,TGS}$ client Ticket Granting Server	~~TGS~~ ~~Usa~~ Client sends TGT and authenticator encrypted with session key and TGS sends back Application server key encrypted with key of App server and a session key, both are encrypted with session key of TGS

0088

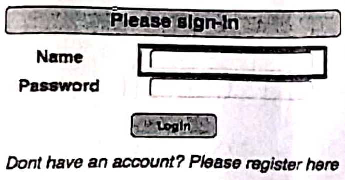


Client sends ticket to S encrypted with key of S and auth encrypted with session key, and ~~[illegible]~~ S sends back timestamp encrypted with public key of client

Question 4: [CLO: 4] [6 marks]

In the given login screen, what input would be required to exploit the SQL injection vulnerability. Moreover, also write down the complete SQL query with your input which results in successful login without valid username and password.

6


Please sign-in

Name

Password

Login

Dont have an account? Please register here

Original Query:
Select * from tablename when name = (input1) and password = (input2)

Input required:
input1 = "John' OR (1=1)--", input2 = "" or any input

Resulting Query:
Select * from table_name when name = 'John' OR (1=1) -- and password = ""

0088

stion 5: [CLO: 4] [3+2+3 marks]

nswer the given questions according to the following code. Note that `scanf()` is used in C to get string/integer input from user. It takes two parameters: `input_format` and `dest_address`. Data in memory is stored in ASCII encoding.

8

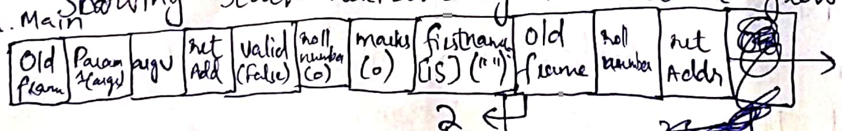
```
int getMarks (int rollNumber);
Boolean checkName (int rollNumber, char* firstname);

int main (int argc, char **argv) ⟶ 1
{
     Boolean valid = False;
     int rollNumber = 0;
     int marks = 0;
     char firstname[15] = "";

     printf("Enter your roll  number: ");
     scanf("%d", &rollNumber);

     // function to get marks from the database
     marks = getMarks(rollNumber); ⟵ 2

     printf("Enter your first name: ");
     scanf("%s", firstname);

     // function ‘checkName’ validates the name of the user against
     // the rollNumber and returns a Boolean value.
     valid = checkName(rollNumber, firstname); ⟶ 3

     if (valid == False)
         return 0;

     printf("%s ", firstname);
     printf("your marks are  %d\n", marks);

     return 0;
}
```

7

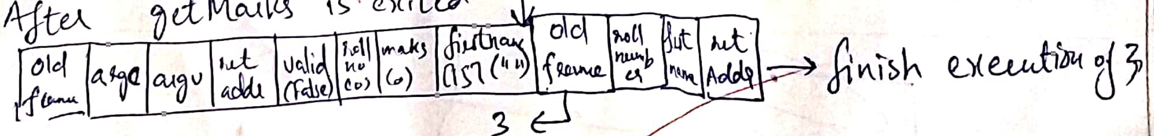
1. **Depict how stack will grow?**

Showing stack horizontally with stack growing towards right

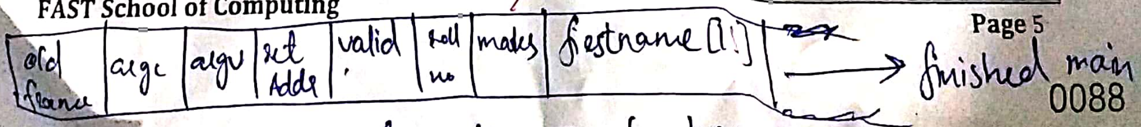
1. Main



After getMarks is exited



After checkname is exited,



After main ends, the stack becomes empty

2. **Identify the problem in the above code?**

The issue is that the scanf is used to take user input in ~~a~~ char array "firstname" with size 15. Since, scanf does not checks size of array and takes input upto null character, if we enter ~~one~~ input of size greater than 15 characters, scanf will not stop and eventually the characters will exceed the size and overwrite adjacent memory ~~abov~~ above/left to "firstname" i.e (marks, rollno, valid, ret addr) So, using scanf() for input is a problem. 2

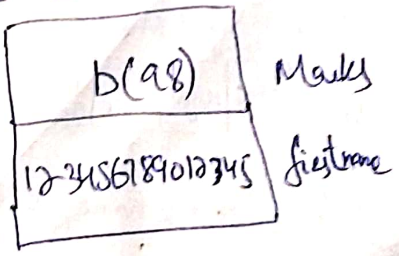
3. **How can you exploit firstname input to display 98 as your marks?**

I will provide firstmame input as ~~"123456789012345~~ "123456789012345b"

Explanation 3

~~Then~~ input is of 16 characters, with 'b' as sixteenth character. Now, since firstname is of 15 characters, the 16th character 'b' will be overwritten in above/left adjacent memory that is of marks. ~~Since~~ Now, the 'b' is of one byte, while marks is of 4 bytes, so 'b' will ocupy first byte of marks, while rest will remain 0. ~~Now~~, since ascii of 'b' is 98 and marks is an int variable, 'b' will be stored as 98 in marks and hence, marks will become 98.

[illegible]


b(98)
Marks
123456789012345
firstname