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

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

4

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. $

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

Question 2: Short Questions

[CLO: 3] [4+2+2

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

8

i. Allow a user 'peter' to create new rows or delete some rows in 'Repository' table. He sh 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 the reading access.

i) GRANT (Insert Rows, Delete Rows} ON [Repository] To [Peter]
REVOKE [ModifyRows, View Table] ON [Repository] FROM [Peter]

ii) GRANT [Reading Access] ON [All Tables] To [Sam]
REVOKE [Writing Access] ON [All Tables] FROM [Sam]
↳ These can be defined as seperate roles in access control.

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.

- Salt randomizes the hash function digest creation.
- Yes, it is important to preserve salts because passwords hashes can be compromised otherwise and hashes can be cracked.

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

- Origin Header is attached with the requests originating from end device and they are matched with target header on the other end.
- Matching the origin and target header will help to detect the malicious requests coming from cross sites.

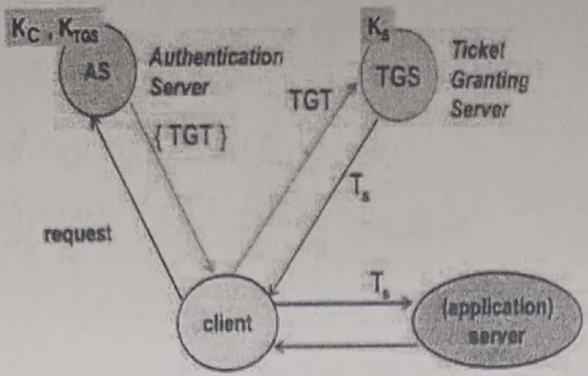
d. What kind of obfuscation techniques is used by retro viruses to avoid detection and analysis?

- Retro viruses masquerade themselves as legitimate files so users will open them.
- Right after the user interaction, the virus gets activated and disturbs the system.
- These self replicating malicious programs will mitigate other systems as well.

3: [CLO: 1] [6 marks]

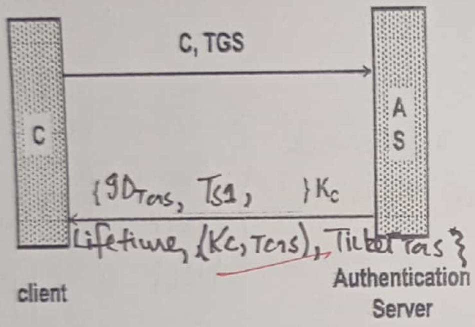
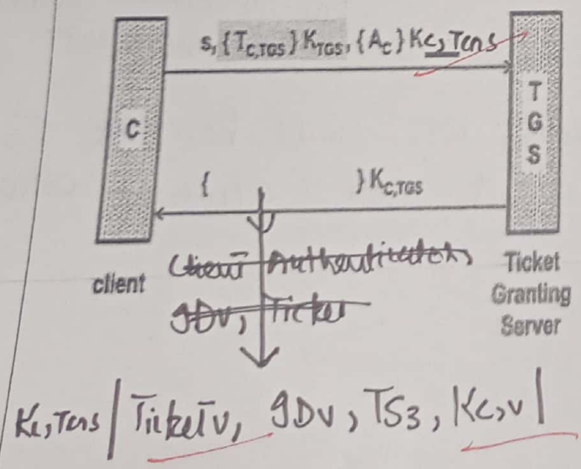
ollowing figure depicts the Kerberos Authentication process with keys involved.

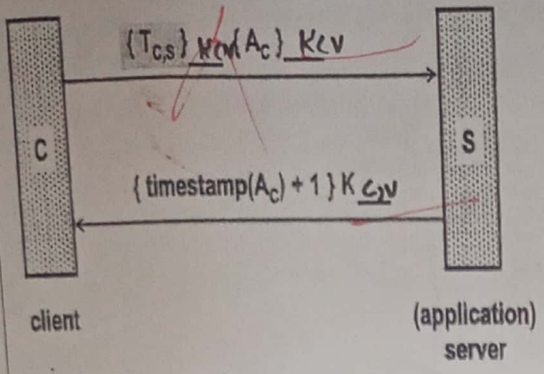
5



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.

. In all cases, I have assumed the server id as V.

 C, TGS C AS {IDTGS, TS1, lifetime, {Kc, TGS}, Ticket TGS } Kc client Authentication Server	• Authentication server gives the client the TGS ticket through which it can contact the TGS • It also provides the session key with this. The whole data is encrypted with the client's key. C ← Kc | IDTGS, TS1, lifetime, Kc,TGS, TicketTGS | AS
 S, {$T_{C,TGS}$} K_{TGS}, {A_C} Kc,TGS C T G S { } $K_{C,TGS}$ ~~Client Authenticator~~ ~~IDv, Ticket~~ client Ticket Granting Server Kc,TGS | TicketV, IDV, TS3, Kc,v |	• Client connects TGS with its authenticator and ticket and server id. • The TGS replies back with the session key, ticket to access a server V, timestamp in the reply. • Through the granted server ticket, the server can be accessed in the session.



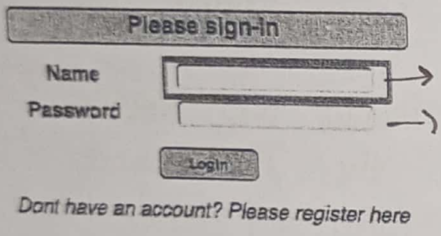
- Client ~~asks~~ contacts the server with its ticket and authentication

- server replies back with the (TS5+1) encrypted with session key (Kc,v).

6

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.



Please sign-in

Name → Any random username

Password → Any random password

} can be left empty as well if no check of empty field is applied.

Login

Dont have an account? Please register here

SQL query:

SELECT FROM users where userid = " " OR 1==1 --

↓ put all other conditions in comment

→ Using logical and boolean operator, we will be able to run the tautology statement which will always return a valid user information.

er the given questions according to the following code. Note that scanf () is used
ng/integer input from user. It takes two parameters: input_format and dest_address.
ta in memory is stored in ASCII encoding.

7

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

int main (int argc, char **argv)
{
    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);

    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);

    if (valid == False)
        return 0;

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

    return 0;
}
```

1. Depict how stack will grow?

~~ret address~~ ✓
~~roll Number~~ ✓
~~main function local variables~~

ret address.	checkname calling.
first name	
roll number	
ret address.	getMarks calling
roll number	
ret address	scanf calling.
destination	
input format	
ret address	printf function calling
printf string	
main function local variables	

:. All printf and scanf will be called in this way.

0132

2. Identify the problem in the above code?

- No input type and range checks have been applied.
- No checks for buffer length are there.
- Users can try exploiting the buffer, integer overflow techniques to disturb the results.

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

Buffer length speified for firstname is 15. We can write 17 characters as the first name input with last two characters as 98. This will cause overflow of name buffer to the marks buffer.