1. What is the difference between Session and Cookie?
2. What is the fundamental cause of SQL Injection Vulnerabilities?
   1. Databases are inherently insecure.
   2. SQL syntax ambiguity.
   3. Mixing code with data.
   4. Issues with the Database structure.
3. Which of the following of these were not mentioned in the lecture as a common countermeasure for SQL Injection?
   1. Filtering and Encoding Data
   2. SQL Spoofing
   3. Prepared Statements
   4. LINQ
4. How does a Prepared Statement protect against SQL Injection?
   1. It sends the code and the data in separate channels to the database server.
   2. It sends the information to the database multiple times.
   3. It prepares the database before the injection occurs.
   4. None of the above.
5. A SQL Injection attack using which of the following SQL statements will allow a user to modify a database?
   1. SELECT
   2. UPDATE
   3. DESCRIBE
   4. USE
6. The following SQL statement is used by a PHP program to send a SQL request to the database, where $eid and $passwd contain data provided by the user in a web form. An attacker wants to try to get Ted’s account information out of the database without knowing his password. What should the attacker put inside the form fields for eid and passwd to achieve that goal. Assume the attacker knows the that Ted’s eid is Ted.

$sql = "SELECT \* FROM employee WHERE eid=’$eid’ and password=’$passwd’"

Your answer should be in the following format, where you follow the equals sign in each case with the string the attacker should type into the web form:

$eid \_\_\_\_\_Ted’ #\_\_\_\_\_\_ $passwd \_\_\_\_\_\_\_Anything\_\_\_\_\_\_\_\_

1. For question 9, what could you do to run an arbitrary SQL statement.

$eid \_\_\_\_\_\_Ted’ or 1=1 #\_\_\_\_\_\_\_\_\_ $passwd \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. The following SQL statement is sent to the database to add a new user to the database, where the content of the $name and $passwd variables are provided by the user, but the EID and Salary field are set by the system. How can a malicious user Bob add himself into the database and set his salary to a value of 1 million?

$sql = “**INSERT INTO** employee (Name, EID, Password, Salary) **VALUES** (‘$name’, ‘EID6000’, ‘$passwd’, 80000)”

Specify what the user should type into the web form fields below to execute the attack.

$name Bob’ , ‘EID0000’ , ‘random’ , 1000000 $passwd \_\_\_\_Anthing\_\_\_\_\_\_\_

1. Explain how a website can use same-site cookie or secret token to prevent CSRF attacks, and why does it

work?

1. Do browsers know whether an HTTP request is cross-site or not? Do servers know whether an HTTP request is cross-site or not? Yes, depending on if they get a cross site token or same site token.
2. Why is it important for a server to know whether a request is cross-site or not? If it is cross site it could be malicious.
3. What are the differences between XSS and CSRF attacks?
4. Can the secret token, same-site cookie countermeasure be used to defeat XSS attacks?

No, the cross site scripting attack is done using a same site token.

1. Cross Site Scripting (XSS) works by:
   1. Causing cross site buffer overflows
   2. Allowing the attacker to insert Javascript code into areas normally meant for textual data.
   3. Elevating the privilege of a cross site user.
   4. None of the above
2. Which of the following is NOT one of the potential damages of Cross Site Scripting that were discussed?
   1. Denial of Service,
   2. Stealing information,
   3. Spoofing requests.
   4. Web defacing (changing the website’s appearance).
3. What are the countermeasures for Cross Site Scripting?
   1. Filtering or Encoding of the user’s HTML inputs.
   2. Encryption.
   3. Prepared Statement.
   4. User Authentication.
4. There is a secret value (int type) stored in the stack. Can you steal this secret value using the Format String Vulnerability? In other words, can you create a format string for the *printf* function to print out this secret value. Assume that the distance between the address of format string and the secret value is 24 Bytes.

A picture containing chart

Description automatically generated

Printf(“”)

1. Which of the following code segments are vulnerable to a Format String Vulnerability (circle all that apply?
   1. scanf(“%s”, user\_input);  
      printf(user\_input);
   2. scanf(“%s”, user\_input);  
      printf(“Numbers %d %d\n”, a, b);
   3. scanf(“%s”, user\_input);  
      printf(user\_input, a, b);
   4. None of the above.