Thapar Institute of Engineering & Technology, Patiala
Department of Computer Science and Engineering

BE (3rd year) EST 11May 2023 UCS638: Secure Coding
Elective Focus: Cyber and Information Security
Time: 03 Hours; MM: 40 Note: 5 questions in total over a spread of 2 pages.

Q1.

i) An organization is concerned about the risk of return to libc attacks on their systems. As a security expert, how would you evaluate the effectiveness of the countermeasures that the organization has implemented to prevent return to libc attacks? Will Threat Modelling help in this scenario, explain STRIDE in this context? **4**

ii) Instead of jump to the system() function, we would like to jump to the execve() function to execute "/bin/sh". Please describe how to do this. You are allowed to have zeros in your input (assume that memcpy() is used for memory copy instead of strcpy())

Q2.

i) What if the SQL statement is constructed in the following way (with a line break in the WHERE clause), can you still launch an effective SQL injection attack? **2**

```
SELECT * FROM employee
WHERE eid= '$eid' AND
password='$password'
```

ii) 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 employee set his/her salary to a value higher than 80000? **2**

```
$sql = "INSERT INTO employee (Name, EID,
Password, Salary) VALUES ('$name', 'EID6000',
'$passwd', 80000)"
```

iii) The following SQL statement is sent to the database, where $eid and $passwd contain data provided by the user. An attacker wants to try to get the database to run an arbitrary SQL statement. What should the attacker put inside $eid or $passwd to achieve that goal. Assume that the database does allow multiple statements to be executed. **2**

```
$sql = "SELECT * FROM employee
WHERE eid='$eid' and
password='$passwd'"
```

iv) One of your classmates without attending secure coding class learnt about prepared statements as one of the deterrent against SQL attacks from the Internet and produced the following code. You attended all the classes and are confident about the subject matter, evaluate the given code carefully and suggest changes if you think code has a vulnerability awaiting to get exploited. **2**

```
$conn = new mysqli("localhost", "root",
"seedubuntu",
"dbtest");
$sql = "SELECT Name, Salary, SSN
FROM employee
WHERE eid= '$eid' and password=?";
if ($stmt = $conn->prepare($sql)) {
$stmt->bind_param("s", $pwd);
$stmt->execute();
...
}
```

Q3.

i) To defeat XSS attacks, a developer decides to implement filtering on the browser side. Basically, the developer plans to add JavaScript code on each page, so before data are sent to the server, it filters out any JavaScript code contained inside the data. Also, developer looked for script tags, and removed them. Let's assume that the filtering logic can be made perfect. Can this approach prevent XSS attacks? **4**

ii) The fundamental cause of XSS vulnerabilities is that HTML allows JavaScript code to be mixed with data. From the security perspective, mixing code with data is very dangerous. XSS gives us an example. Provide two other examples that can be used to demonstrate that mixing code with data is bad for security. Just naming will not attract any credit, give example code/scenarios to justify your answer.