Experiment No. 7

Title: Challenge-Response Protocol

Batch: B2 **Roll No.: 16010421119** **Experiment No.: 7**

Title: Design and implement a VLab for Challenge-Response protocol.

Resources needed: Windows/Linux OS

Theory:

Pre Lab/ Prior Concepts:

Consider a situation where a server (for example, a base station) wants to authenticate a client (a mobile phone user) by confirming that the client has the correct password (say, a 5-digit password PSWD).

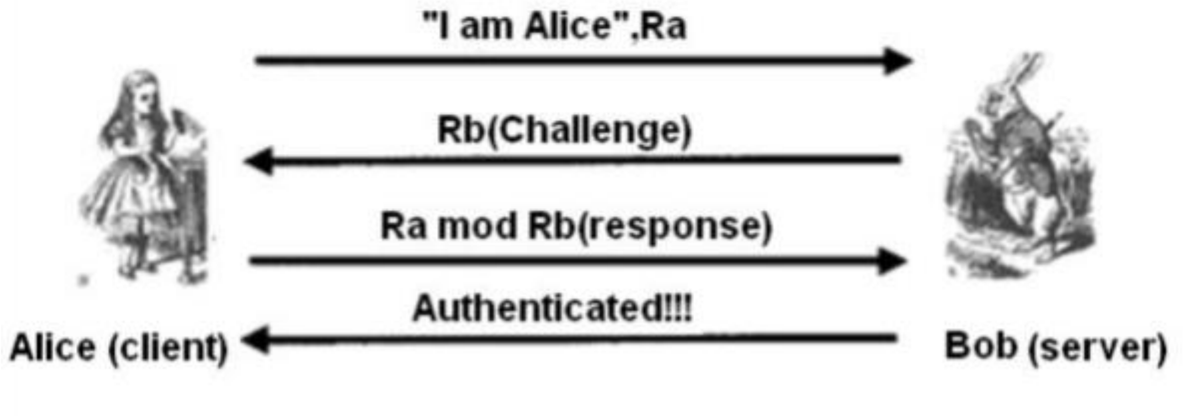
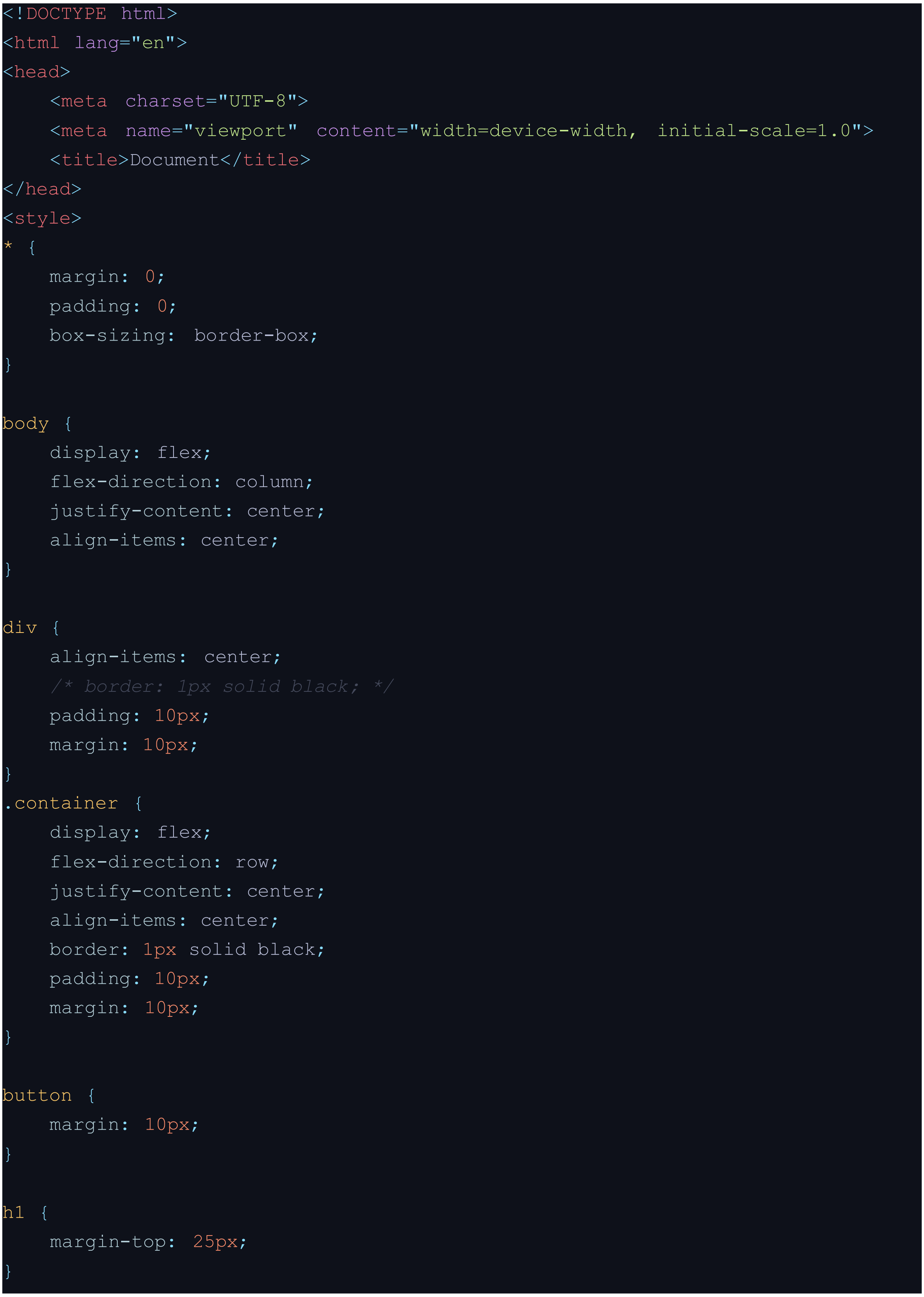


Figure - Challange-Response Protocol

Assume there are malicious eaves-droppers who can hear the communication that is taking place. A simple authentication method is as follows: The server generates a random 3-digit number RAND and sends it to the client. The client computes the remainder(PSWD mod RAND) and sends the result to the server. The server also computes the value (PSWD mod RAND) and if it gets the same result, it concludes that the client has the correct password and authenticates the client as shown in figure 1.

Procedure / Approach /Algorithm / Activity Diagram:



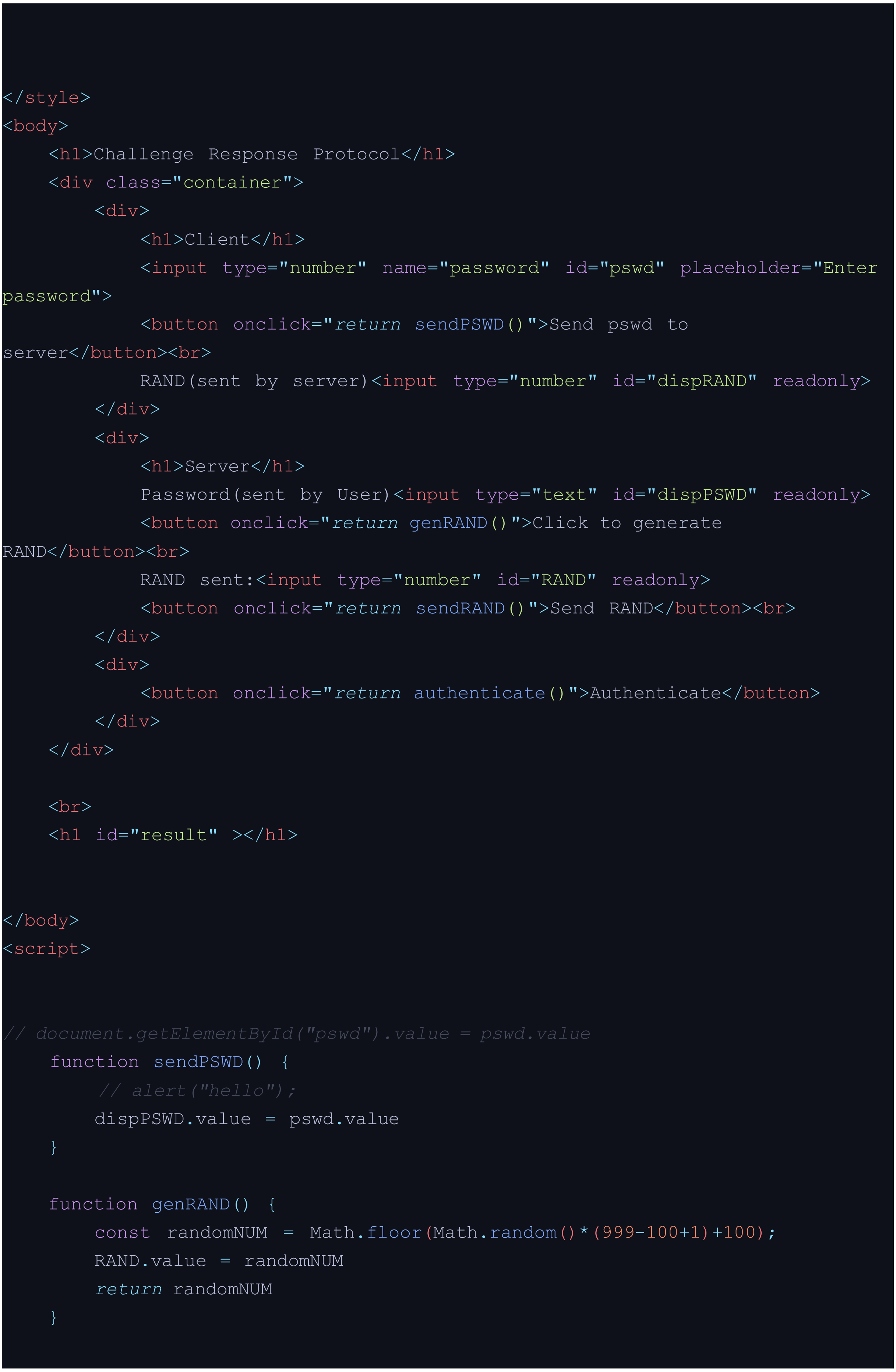
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<style>
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

body {
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
}

div {
    align-items: center;
    /* border: 1px solid black; */
    padding: 10px;
    margin: 10px;
}
.container {
    display: flex;
    flex-direction: row;
    justify-content: center;
    align-items: center;
    border: 1px solid black;
    padding: 10px;
    margin: 10px;
}

button {
    margin: 10px;
}

h1 {
    margin-top: 25px;
}
```



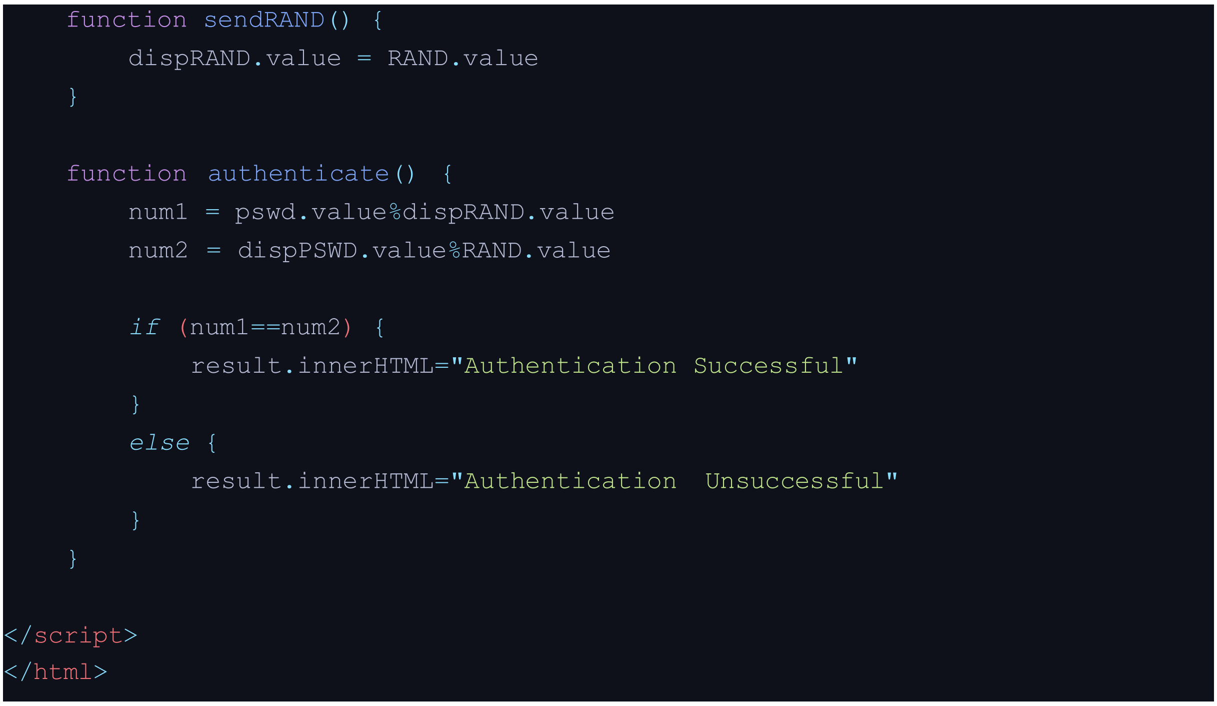
```
</style>
<body>
    <h1>Challenge Response Protocol</h1>
    <div class="container">
        <div>
            <h1>Client</h1>
            <input type="number" name="password" id="pswd" placeholder="Enter
password">
            <button onclick="return sendPSWD()">Send pswd to
server</button><br>
            RAND(sent by server)<input type="number" id="dispRAND" readonly>
        </div>
        <div>
            <h1>Server</h1>
            Password(sent by User)<input type="text" id="dispPSWD" readonly>
            <button onclick="return genRAND()">Click to generate
RAND</button><br>
            RAND sent:<input type="number" id="RAND" readonly>
            <button onclick="return sendRAND()">Send RAND</button><br>
        </div>
        <div>
            <button onclick="return authenticate()">Authenticate</button>
        </div>
    </div>

    <br>
    <h1 id="result" ></h1>


</body>
<script>


// document.getElementById("pswd").value = pswd.value
    function sendPSWD() {
        // alert("hello");
        dispPSWD.value = pswd.value
    }

    function genRAND() {
        const randomNUM = Math.floor(Math.random()*(999-100+1)+100);
        RAND.value = randomNUM
        return randomNUM
    }
```



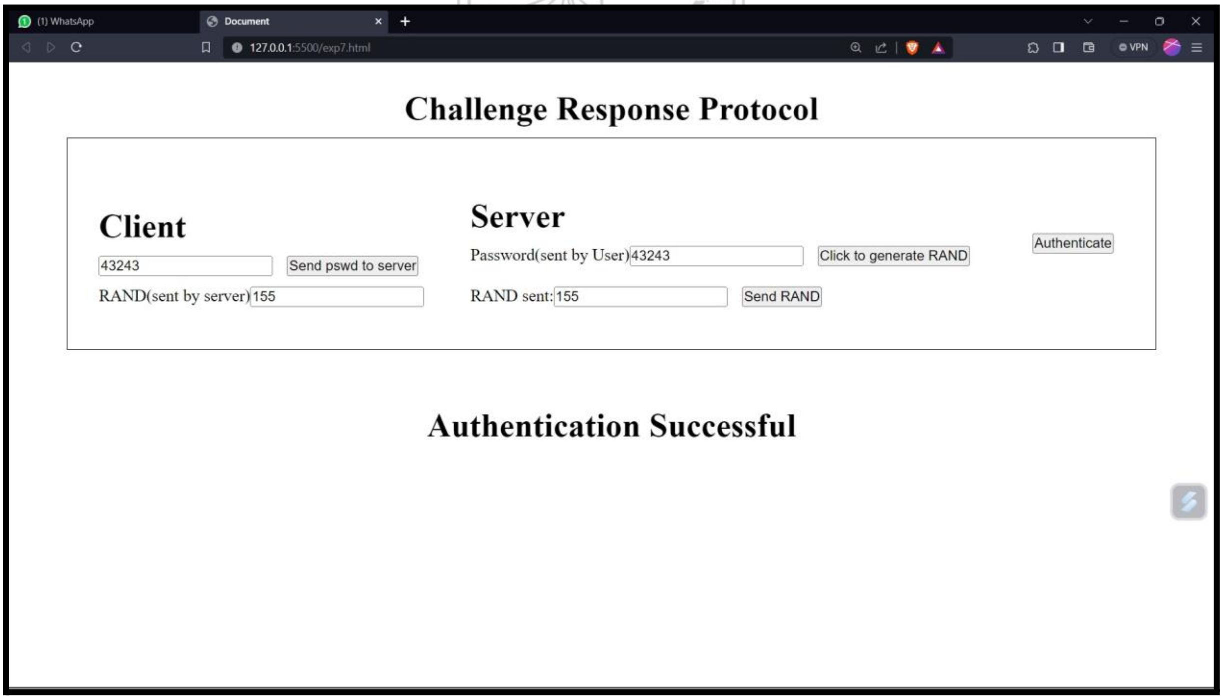
```
    function sendRAND() {
        dispRAND.value = RAND.value
    }

    function authenticate() {
        num1 = pswd.value%dispRAND.value
        num2 = dispPSWD.value%RAND.value

        if (num1==num2) {
            result.innerHTML="Authentication Successful"
        }
        else {
            result.innerHTML="Authentication  Unsuccessful"
        }
    }

</script>
</html>
```

Output:



Challenge Response Protocol

Client

43243 | Send pswd to server

RAND(sent by server) 155

Server

Password(sent by User) 43243 | Click to generate RAND | Authenticate

RAND sent: 155 | Send RAND

Authentication Successful

Questions:

1. What are the advantages and disadvantages of the above authentication method?
Ans.
Advantages:

Enhanced Security: Challenge-response protocols can provide a higher level of security compared to simple password-based authentication. They require the user to possess something (knowledge of a secret or possession of a token) and prove it by responding correctly to a challenge.

Resistance to Replay Attacks: Challenge-response systems often incorporate elements to prevent replay attacks, where an attacker intercepts and reuses a previous authentication session. The challenge is typically generated in a way that makes it difficult to predict or reuse.

Two-Factor Authentication (2FA): Challenge-response methods can easily be combined with other authentication factors, such as something the user knows (e.g., a PIN) and something the user has (e.g., a physical token or smartphone). This adds an additional layer of security.

Reduced Password Exposure: Unlike traditional password-based systems where a password is stored on a server and can be a target for hackers, challenge-response systems do not store the secret in a reversible form, making it harder for attackers to obtain user credentials.

Versatility: Challenge-response protocols can be adapted to various scenarios and security requirements. They are not limited to specific authentication mechanisms and can be used in both online and offline environments.

Disadvantages:

Usability: Challenge-response authentication methods can be less user-friendly than traditional password-based methods. Users may find the process of responding to challenges cumbersome, especially if it involves additional hardware tokens or complex procedures.

Cost: Implementing challenge-response systems may require the deployment of additional hardware tokens or mobile apps, which can increase the cost of authentication infrastructure.

Lockout Risk: If a user fails to respond correctly to a challenge multiple times (due to forgetfulness or other reasons), they may be locked out of their account, leading to usability issues and the need for account recovery procedures.

Phishing Vulnerability: While challenge-response methods can be resistant to password-based attacks, they are not immune to phishing attacks. Attackers can still trick users into providing their responses to fraudulent challenges.

2. Explain replay attack on this protocol?

Ans. In the context of challenge-response protocol authentication methods, a replay attack occurs when an attacker intercepts the challenge and response exchanged between the user and the system and then attempts to replay (retransmit) them to gain unauthorized access or perform malicious actions.

Initiation: The legitimate user initiates the authentication process by sending a request to the system they want to access.

Challenge Generation: The system responds by generating a random challenge, which is a one-time piece of data used to verify the user's identity.

User Response: The user receives the challenge and responds with the appropriate response, which is typically based on a secret or cryptographic key and the challenge.

Successful Authentication: The system verifies the user's response, and if it matches the expected value, the user is granted access.

Outcomes:

CO3: *Describe various access control policies and models*

Conclusion:

Learnt and implement a virtual lab on the concept of authentication using challenge response protocol

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of faculty in-charge with date

References:

Books/ Journals/ Websites:

- Mark Stamp, “Information security Principles and Practice” Wiley.