**A Security Plan for a secure Web Page**

**Form- Data Given:**

Username

Password

Credit Card Information

**Methods included to keep the data secure**

**Avoid SQL Injection:**

Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker’s hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.

Let’s understand this from form data viewpoint.

If a form is using get method the form data will be exposed to unintended user at the data will be appended to URL while using a GET request.

However using a POST request will avoid such scenario as the POST method does not append the data to the URL.

So if a page is using a form and it is using get method to transmit the data, there is a great possibility that the data can be exploited by unintended users.

Action taken : Use of POST method to submit the form data.

Form Data Affected : Username , Password, Credit Card Information.

Advantage : No sensitive Data exposure.

**Avoid Broken Authentication:**

Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users’ identities temporarily or permanently.

So if there is no form-validation present on the webpage, users can store any value in the database and it will be executed and unintended may get transferred to others.

Also if the page is logging in there must be a logout page in order to destroy the session and session data. Keeping session data active for very long time may get users into trouble as malicious codes running out in the web are continuously searching for such scenarios.

Action Taken : Proper login/logout, Proper form validation.

Form Data Affected : Username , Password, Credit Card Information.

Advantage : No sensitive Data exposure, No common username/password combination allowed.

**Encryption of Sensitive Data:**

Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser.

Action Taken : Encryption methods used.

Form Data Affected : Password, Credit Card Information.

Advantage : No sensitive Data exposure.

**Properly Configured Headers**

Headers like content-security-policy , referrer-policy , permission policy , X-Content-type options keep websites away from any vulnerable attack. Case in point being where a malicious code present in some link on the page and if X-Content-Type header is not configured properly, the page will execute it without checking the mime or anything.

Browser sniffing does not always prove to be good so including X-Content-Type header will remove any chances of execution of any unintended code present on the page.

Action Taken : Headers configured properly.

Form Data Affected : Username , Password, Credit Card Information.

Advantage : No sensitive Data exposure, No unwanted code execution.