Harsh Sharma
Karan Saini

Akash Anand
Tushar Gupta

Common Vulnerability Attacks as per OWASP

1. **Injection**:

Injection flaws, such as SQL injection, LDAP injection, and CRLF injection, occur when an attacker sends untrusted data to an interpreter that is executed as a command without proper authorization.

2. **Broken Authentication and Session Management:**

Incorrectly configured user and session authentication could allow attackers to compromise passwords, keys, or session tokens, or take control of users' accounts to assume their identities.

3. **Sensitive Data Exposure**:

Applications and APIs that don't properly protect sensitive data such as financial data, usernames and passwords, or health information, could enable attackers to access such information to commit fraud or steal identities.

4. **Broken Access Control**

Improperly configured or missing restrictions on authenticated users allow them to access unauthorized functionality or data, such as accessing other users' accounts, viewing sensitive documents, and modifying data and access rights.

5. **Security Misconfiguration**

This risk refers to improper implementation of controls intended to keep application data safe, such as misconfiguration of security headers, error messages containing sensitive information (information leakage), and not patching or upgrading systems, frameworks, and components.

Tested the Web App for the following Threats without implementing WAF

The following vulnerabilities were detected for the Web App:
Kali Tool Used : Vega

Vulnerability Test 1
Attack Vector -> SQL Injection
Risk Type: High

Result:
We detected a possible SQL injection vulnerability. These vulnerabilities are present when externally-supplied input is used to construct a SQL query. If precautions are not taken, the externally-supplied input (usually a GET or POST parameter) can modify the query string such that it performs unintended actions. These actions include gaining unauthorized read or write access to the data stored in the database, as well as modifying the logic of the application.

Classification	Input Validation Error
Resource	https://www.csye6225-fall2018-sharmaha.me/login
Parameter	username
Method	POST
Detection Type	Blind Text Injection Differential
Risk	High

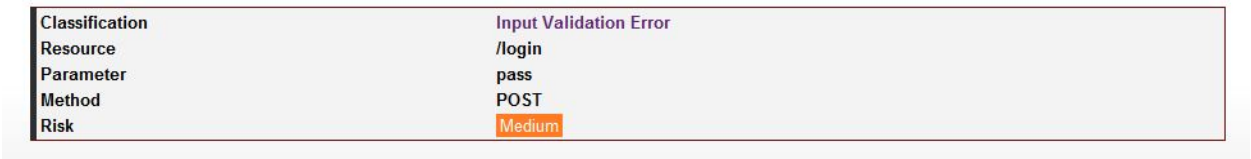
Without implementing Firewall

Impact:

We detected a possible SQL injection vulnerability.
These vulnerabilities can be exploited by remote attackers to gain unauthorized read or write access to the underlying database.
Exploitation of SQL injection vulnerabilities can also allow for attacks against the logic of the application.
Attackers may be able to obtain unauthorized access to the server hosting the database.

Vulnerability Test 2
Attack Vector -> XSS - XML Injection
Risk Type: Medium
Result:
XML injection can occur when externally supplied data that has not been sufficiently validated is used to create an XML document. It is possible for this data to corrupt the structure of the documents. The

possible consequences depend on the XML document and what it is used for.



Classification	Input Validation Error
Resource	/login
Parameter	pass
Method	POST
Risk	Medium

Vulnerability Test 3

Attack Vector: Broken Authentication and Session Management:
Risk Type: Medium
Vega has detected that a known session cookie may have been set without the secure flag.

Impact

Cookies can be exposed to network eavesdroppers.
Session cookies are authentication credentials; attackers who obtain them can get unauthorized access to affected web applications.

Manual Testing

1. IP blacklisting
   Tested by adding a random ip to IPMatch Set and bloked when the firewall was implemented.
   The ip was blocked access to the web app.

2. File Restriction
   File restriction upload set to 100KB and test successfully using postman.