Lab 9: Technical Risk Analysis and Static Analysis

Risk ID	Technical Risk	Technical Risk Indicators	Related CWE or CVE ID	Impact Rating	Impact	Mitigation	Validation Steps
1	**Cross-Site Scripting (XSS)**	There are some JavaScript functions running that were not the part of given application thus making the site appear to users different than it really should be.	CWE-79 https://cwe.mitre.org/data/definitions/79.html	M	It allows the intruder to embed malicious content on the website and may even make the website extremely unresponsive for the authentic users.	System should check for valid input, filter out words and characters that indicate the presence of injected script and sanitize the output from the user input	An attempt to inject JavaScript code instead of valid data and ensure that the site's original content remains unmodified resulting this attempt of XSS.
2	**User authentication system can be brute-forced**	Failed continuous login attempts are not prevented.	CWE-307 https://cwe.mitre.org/data/definitions/307.html	H	Allows the attacker to bypass the login form and get access to secure information.	Locking the system after specific number of failed attempts to login such as locking the system for half an hour after 5 unsuccessful login attempts and owner to be notified.	Freeze the input credentials form. This can be done by locking the submit form button.
3	**Source code in .git directories publicly accessible**	Allows public access to git directories that contained the flag.txt without any authentication. These files can be accessed	CWE-538 https://cwe.mitre.org/data/definitions/538.html	H	Attackers can use the mistakenly publicly available site's source code to understand the way system works and hence can use malicious ways to attack and damage	Restrict access to all the files containing the source code by using commands such as 'chmod'.	.git directory is no longer accessible using the directory traversal in URL.

		using directory traversal in URL.			the server in any way they want.		
4	**SQL Injection**	Possible to bypass the login using “a‘ or ‘1’=’1” as both the password and username that sets the condition to true. Also, it is possible to access the database of the server by injection of SQL commands in URL i.e. id parameter of posts in the URL was injectable.	CWE-89 https://cwe.mitre.org/data/definitions/89.html	H	Allows the hacker to get invalid access to the application and view the sensitive information stored in the database such as the user credentials including the password hashes file.	Setup two-way verification method for login and use prepared statements for communicating with SQL server rather than using dynamically creating SQL queries. Also, make sure that the user data in both the URL and input forms is validated.	Try SQL injection to gain access to the application and maybe use tools such as sqlmap to make sure that none of the parameters in the URL are injectable.
5	**Cookie Tampering**	It is possible to access and alter the cookie using the browser’s web tools.	CWE-565 https://cwe.mitre.org/data/definitions/565.html	H	Hackers can modify the cookie by setting the admin flag to true and get access to the pages which, technically, should only be accessed by the admin.	Encrypt the information stored in the cookies.	The encrypted information in the cookies is meaningless and hence defensive against any alterations.

6	**Hardcoded Passwords in php files**	Usernames and passwords are hard coded and can be found in the dblib.php file.	CWE-798 https://cwe.mitre.org/data/definitions/798.html	H	An account can be compromised if a hacker finds a hardcoded password. Such a vulnerability exploitation needs restructuring of the whole system.	Passwords should not be stored in the source code files of the application. These passwords should be stores in a secure location with additional layers of validation and each of the passwords should be hashed for extra security.	Ensure that passwords are no longer stored in the application's source code files but rather are stores in a secure database with the hash feature enabled.
7	**Directory traversal allows public access to some important information**	It is possible to traverse and view the directories in the in the URL.	CWE-23 https://cwe.mitre.org/data/definitions/23.html	M	Allows the attacker to get access to sensitive information present in directories that provide the path to other viewable files without any authentication, and thus important information can be leaked.	Restrict the view to all directories that provide the path to other viewable files but themselves does not need to be shown.	The permissions have been set on all appropriate directories and files, and thus directories are no longer visible after attempts of traversal in the URL.