Risk ID	Technical Risk	Technical Risk Indicators	Related CVE, CWE, OSVDB IDs	Impact Rating	Impact	Mitigation	Validation Steps
1	SQL Injection	SQL code added through query parameters is executed. (board.php 55, 61; dblib.php 23)	CWE-89	H	Any information stored in the SQL database can be retrieved without authorization.	Validate user-supplied input, or properly escape special characters like quotes.	Attempted SQL injections are rejected, or no match is found for the query.
2	Cross-Site-Scripting	JavaScript code submitted to message board is executed. (board.php 43, 58)	CWE-80	H	Attackers can run malicious scripts on other users' machines through the message board.	Validate user-supplied input, or properly escape special HTML characters before displaying to other users.	Board submissions with HTML tags are rejected, or are displayed literally instead of being executed.
3	Improper access control	SQL queries are executed as the root user, in a database including sensitive user information. (board.php, dblib.php)	CWE-284, CWE-266	M	Along with the SQL injection vulnerability, attackers can gain access to all information in the database, which include usernames and password hashes.	Use different SQL users to access different tables. Access the message board with a low-permission user, so such queries cannot access more sensitive parts of the database, like user information.	User information is not accessible through user queries.
4	Directory Listing	Directory listings are sent to the user, potentially exposing information.	CWE-548	M	Users can navigate the files in a number of directories that are not meant to be accessible, potentially revealing confidential files and giving attackers knowledge about the directory structure of the web site.	Disable directory listing in the server's configuration.	Navigation to wp-includes/ and subdirectories results in a 404 error.

Risk ID	Technical Risk	Technical Risk Indicators	Related CVE, CWE, OSVDB IDs	Impact Rating	Impact	Mitigation	Validation Steps
5	Passwords not salted	Hashes for passwords are calculated without a salt, so users with the same password have the same hash stored in the database. (dblib.php 23)	CWE-759, CVE-2006-1058	M	Attackers with access to users and password hashes can more easily find passwords, both by precomputing them, and by looking for repeated hashes.	Assign a salt to each password when it is created, and store this in the database. Append this to the password before hashing.	There are no repeat hashes in the password database.
6	Anony-mous FTP enabled	Users can connect to the server through FTP.	CVE-1999-0497	M	Unauthorized users can connect to the server through FTP, exposing information to unauthorized users.	Disable anonymous FTP.	Attempts to connect to the FTP port without a login fail.
7	Error message exposure	Errors in SQL database are reported directly to user. (dblib.php 8, 25-7; board.php 18; scoreboard/index.php 34, 114)	CWE-209	L	Error message gives information about how SQL is being used, which can make it easier for attackers to find weaknesses.	Do not display error messages to users. Either direct users to a vague error page, respond with a 500 error, or deal with the errors on the server side (eg. treating a SQL error as if no match was found for the query).	Error messages are not shown to the user.
8	Weak password require-ments	Some users have very weak passwords.	CWE-521	L	Attackers can easily compromise accounts of users with weak passwords.	Require a certain strength level on passwords when they are created/changed.	A password cracker run on the database of users fails to find passwords within a fixed timeframe (eg. a minute).