Risk ID	Technical Risk	Technical Risk Indicators	Related CWE ID	Impact Rating	Impact	Mitigation	Validation Steps	Sources
1	Obvious administrative username: root	Root is the username set for the username of the wordpress blog	CWE-341,	Medium	Attackers can bruteforce the password for root and gain admin access	Set the wordpress admin username to anything other than root	User name is not as easily guessed by crackers.	CWE
2	Response Discrepancy Information Exposure: webapp will say if user is valid	Admin log-in tells whether failed log-in attempts are due to invalid username or bad password	CWE 204	Medium	Attackers can bruteforce for valid usernames then crack passwords conventionally	Only specify if log-in attempt failed or not without providing additional information	accounts are secure and usernames are not being easily guessed	CWE
3	Informtion Exposure through Discrepancy, attempt to connect to FTP results in key	FTP port is open and can be scanned by Nmap	CWE-203	Medium	Loss of key	Close FTP port and don't respond to it being connected to	FTP port remains closed	CWE
4	Sql Injection allows unauthorized access to database	Sql code embedded in normal queries; unauthorized changes to database	CWE 89 CVE-2008-5817	high	loss of sensitve information such as passwords/keys	Sanitize all input in query fields, filtering out special characters	No changes in database and only authorized queries occur	Veracode, CWE
5	Keys/passwords are hard-coded into certain paths of the webapp	fuzzing or attempting to accesss the wp-uploads directory of the webapp reveals sensitive information	CWE 259	High	Easy access to sensitive information	Store passwords in non-public locations; never store as plain text	No passwords can be accessed simply by navigating the website	Veracode, CWE

6	Cross-site scripting through user-input of HTML script tags	Users are able to manipulate the content of the webapp	CWE 80	High	Redirection to malicious pages or vandalism of web-page	Sanitize user-input, filter out all HTML tags from user input.	Only intended changes to the webapp can be seen; Users can only post text.	Veracode, CWE
7	Plaintext storage of password in memory	Anyone with access to the system can easily find the unencrypted passwords	CWE 316	Medium	loss of sensitve information such as passwords/keys	Don't store plaintext passwords. Always hash/encrypt it	Can't find plaintext sensitive information in memory	Veracode, CWE
8	Lack of encryption for Sensitive data	Certain functions may be passed unencrypted data	CWE 311	Medium	Unintended sharing of sensitive data	Make sure all functions receive sensitive data encrypted	Web app is not passing sensitve data around un encrypted	Veracode, CWE
9	Use of Broken Cryptographic Algorithm	Sensitive data is easily cracked using conventional mehtods	CWE 327	Medium	Passwords/keys are easily obtainable	Use ligitimate/trusted crytographic algorithms	Passwords and Keys cannot be easily cracked normally	Veracode, CWE
10	User has access to directory containing contnents of app	user of app can traverse to directories unintended by the creator and access data	CWE 73	Medium	Easy unauthorized access to other directories by users	Ensure user cannot get to directory listing thorugh URL	Only intended public webpages are accessible by Users	Veracode, CWE
11	Exposure of Information thorugh Error message	error message provides more information than intended	CWE 209	Low	Sensitive information can be leaked	Ensure error message only describes the error itself	Error messages only give information on current issure	Veracode, CWE

12	External Initialization of variables/Data	Optarg is unbounded and attackers can overflow the destination buffer	CWE 454	Low	Can result in execution of unwanted code	Limit size of data copied form optarg variable	Command line applications are only run when intended by the programmer	Veracode, CWE