Material included was either found by inspection of code, analyzing the website for vulnerabilities, or through static analysis by Veracode (confirmed by inspection of code in the ctf-2015/www folder to ensure findings had a legitimate risk of exposing information or providing an attack vector). I did not analyze code in wp-admin, wp-content, and wp-includes directories or other wp files for risks.							
Risk ID	**Technical Risk**	**Technical Risk Indicators**	**CVE, CWE, OSVDB IDs**	**Impact Rating**	**Impact**	**Mitigation**	**Validation Steps**
1	SQL Injection - arbitrary database information may be accessed by supplying carefully crafted input to id field in a request to board.php	Database logs may show unexpected commands run. In severe cases, database tables may have been dropped, causing errors when attempting to access those tables.	CWE 89	H	Sensitive information can be accessed, compromising security. Information may be lost in attacks that cause tables to be dropped.	Perform validation on the input directly before using in SQL commands (server side validation).	User input that contains a SQL injection that is passed to the database has characters such as single quotations escaped.
2	SQL Injection - log in to an account without a password by supplying carefully crafted input to username field in includes/dblib.php (input occurs on login.php page)	Database logs may show unexpected commands run. In severe cases, database tables may have been dropped, causing errors when attempting to access those tables.	CWE 89	H	Sensitive information can be accessed, compromising security. Information may be lost in attacks that cause tables to be dropped.	Perform validation on the input directly before using in SQL commands (server side validation).	User input that contains a SQL injection that is passed to the database has characters such as single quotations escaped.
3	Hard-coded credentials in source code in board.php	Log in credentials for a service such as a database appear in plaintext in code when page source is viewed. Unauthorized log ins to the service may occur, which may result in settings or information being changed, or becoming locked out of the account due to credentials being changed.	CWE 259	H	An attacker may be able to use those credentials to gain access to sensitive information or make unauthorized requests. If this is for an api or service that charges, the information may be used to access that service to an extent that causes a large expense to the owner of the account. Finally, access to the account may be lost should an attacker be able to change the credentials once account access has been gained.	Refactor code so that instead of credentials appearing in plain text, they are stored somewhere secure, such as environment variables on the server.	Account credentials should no longer be found in plaintext if source code is viewed.
4	Hard-coded credentials in source code in scoreboard/index.php	Log in credentials for a service such as a database appear in plaintext in code when page source is viewed. Unauthorized log ins to the service may occur, which may result in settings or information being changed, or becoming locked out of the account due to credentials being changed.	CWE 259	H	An attacker may be able to use those credentials to gain access to sensitive information or make unauthorized requests. If this is for an api or service that charges, the information may be used to access that service to an extent that causes a large expense to the owner of the account. Finally, access to the account may be lost should an attacker be able to change the credentials once account access has been gained.	Refactor code so that instead of credentials appearing in plain text, they are stored somewhere secure, such as environment variables on the server.	Account credentials should no longer be found in plaintext if source code is viewed.
5	Hard-coded credentials in source code in includes/dblib.php	Log in credentials for a service such as a database appear in plaintext when source code. While this file is not ever displayed to a browser, if it is in a public repository the credentials are still available. Unauthorized log ins to the service may occur, which may result in settings or information being changed, or becoming locked out of the account due to credentials being changed.	CWE 259	H	An attacker may be able to use those credentials to gain access to sensitive information or make unauthorized requests. If this is for an api or service that charges, the information may be used to access that service to an extent that causes a large expense to the owner of the account. Finally, access to the account may be lost should an attacker be able to change the credentials once account access has been gained.	Refactor code so that instead of credentials appearing in plain text, they are stored somewhere secure, such as environment variables on the server.	Account credentials should no longer be found in plaintext if source code is viewed.
6	Cross-Site Scripting/XSS - user input displayed on the page may run user defined scripts included in the input in the title field for making a post in board.php	A page that displays user input as content may change its appearance or content. This may include different text or images, changes to a form, or even redirecting to another page.	CWE 80	H	An attacker can take business away from a site by redirecting to another. In more severe cases, a log in form may be replaced with one that will send the credentials to the attacker, compromising many accounts.	Perform validation on the input on server side to escape or strip out <script> tags before the input is stored or displayed.	When content including <script> tags is entered and displayed, instead of making possibly malicious changes to the page, the escaped script tags and the script will be visible in the displayed content.
7	Cross-Site Scripting/XSS - user input displayed on the page may run user defined scripts included in the input in the post field for making a post in board.php	A page that displays user input as content may change its appearance or content. This may include different text or images, changes to a form, or even redirecting to another page.	CWE 80	H	An attacker can take business away from a site by redirecting to another. In more severe cases, a log in form may be replaced with one that will send the credentials to the attacker, compromising many accounts.	Perform validation on the input on server side to escape or strip out <script> tags before the input is stored or displayed.	When content including <script> tags is entered and displayed, instead of making possibly malicious changes to the page, the escaped script tags and the script will be visible in the displayed content.

8	Cross-Site Scripting/XSS - user input displayed on the page may run user defined scripts included in the input in the comments field for replying to a post in board.php	A page that displays user input as content may change its appearance or content. This may include different text or images, changes to a form, or even redirecting to another page.	CWE 80	H	An attacker can take business away from a site by redirecting to another. In more severe cases, a log in form may be replaced with one that will send the credentials to the attacker, compromising many accounts.	Perform validation on the input on server side to escape or strip out <script> tags before the input is stored or displayed.	When content including <script> tags is entered and displayed, instead of making possibly malicious changes to the page, the escaped script tags and the script will be visible in the displayed content.
9	Missing encryption of sensitive data (Veracode) traveling over the network for login request in admin.php	Successful requests containing sensitive information are made with HTTP instead of HTTPS.	CWE 311	H	Cryptographic keys or similar sensitive data may be exposed to an attacker when it travels over a network.	When sensitive information is passed to functions, ensure that it is encrypted, such as by using SSL.	Sensitive information is no longer visible in plaintext as it travels over the network.
10	Use of a broken or risky cryptographic algorithm (Veracode) - SHA1 in password encryption in includes/dblib.php	Hashes for the passwords in the database may appear in a rainbow tables.	CWE 327	M	Sensitive information may be accessed through known ways of breaking the algorithm, for example accounts may be accessed due to hash collisions for passwords.	Change the algorithm used to one that is known to be secure, for the moment at least.	
11	Possible information exposure through an error message (Veracode) in board.php, specifically when connecting to the database	Focused attacks that show knowledge of the database that could be given in SQL error message are occuring.	CWE 209	L	Attackers may be able to use information provided in error messages, such as stack traces, to make more focused attacks.	Refactor error messages provided to the end user to be generic rather than providing information about the implementation or contain information such as passwords	Supplying input that should result in an error should display only generic information about the error to the user.
12	Possible information exposure through an error message (Veracode) in scoreboard/index.php, specifically when connecting to the database	Focused attacks that show knowledge of the database that could be given in SQL error message are occuring.	CWE 209	L	Attackers may be able to use information provided in error messages, such as stack traces, to make more focused attacks.	Refactor error messages provided to the end user to be generic rather than providing information about the implementation or contain information such as passwords	Supplying input that should result in an error should display only generic information about the error to the user.
13	Possible information exposure through an error message (Veracode) in includes/dblib.php, specifically when connecting to the database	Focused attacks that show knowledge of the database that could be given in SQL error message are occuring.	CWE 209	L	Attackers may be able to use information provided in error messages, such as stack traces, to make more focused attacks.	Refactor error messages provided to the end user to be generic rather than providing information about the implementation or contain information such as passwords	Supplying input that should result in an error should display only generic information about the error to the user.
14	Information exposure through directory listing (CWE) for /wp-content/uploads and subdirectories	Server logs show successful access to files that should be restricted against access by users.	CWE 548	M	An attacker may access a directory listing, and therefore files meant to be private, by using parts of URLs used to access permitted resources.	Ensure that privacy settings on files and directories correctly restrict access and that the server is not permitted to automatically generate directory listings.	URLs that formerly resulted in directory listings are now invalid.
15	Use of a One-Way Hash without a Salt (CWE) in includes/dblib.php makes leaked password hashes more vulnerable.	If the password hashes are exposed, passwords will be cracked and exploited fairly quickly after the exposure.	CWE 759	M	The lack of a salt increases the spead at which passwords may be cracked because the contents of a dictionary may be pre-computed.	Create salts when accounts are created and add it to the password before hashing, saving the salt along with the hashed password in the account information. Append the salt to entered passwords before hashing and checking against the saved hash.	Running a dictionary attack with precomputed hashes against a list of the hashed passwords should not yield any results.
16	Login can be acheived through brute force	Many login attempts to same user may appear in server logs.	CWE 307	H	Attackers can gain access to accounts by guessing credentials.	Set a limit to invalid logins in a row before account is locked, and keep track in the database.	If more than the allowed attempts have been reached, an error is displayed and it is reflected in the database.
17	Outdated version of WordPress is in use	Navigating to the readme page shows an older version of WordPress than the newest		M	Attackers may use exploits and vulnerabilities to access information that may have since be patched. The level pf danger depends on what the software is and what has been fixed in newer versions.	Update the version of WordPress to the most current.	The readme should now show the most current version.