Risk ID	Technical Risk	Technical Risk Indicators	Impact Rating	Impact	Mitigation	Validation Steps
1	Eval Injection	Allows user input to the fed directly into a function (eval) the executes the input as code. Eval() used in code.	VH	Malicious code can be ejected and executed by the system resulting in data loss or corruption, lack of accountability, or denial of access.	Validate all user input to ensure it conforms to the expected format. Avoid executing code derived from untrusted input	Do not execute code that comes from an untrustworthy source
2	PHP Remote file inclusion	PHP application receives user input but does not properly restrict the input before using it in require(), include() and similar functions	H	Attacker can specify a URL to a remote location from which the application will retreive code and exeute it. Comprimises site and useability.	Validate all user input to ensure it conforms to the expected format. Avoid executing code derived from untrusted input	Do not allow untrusted input to be evaluated or otherwise interpreted as code
3	SQL Injection	Data in the databaes has been accessed, modified, or deleted from an attacker.	H	Allows attacker to execute arbitrary SQL queries against the database and gain access to the filesystem and perform administrative operations on the database. Data could be lost, stolen, or changed.	Use parameterized prepared statements rather than dynamically constructing SQL queries	Do not directly execute code that comes from a user and be sure to use prepared statements.
4	Use of Hard-Coded Password	Passwords can be easily compromised	M	Account being protected can be compromised and the entire software must be patched to change the password. All deployed instances are vulnerable.	Store passwords out of band from the application code	Follow best practices for protecting credentials stored in locations such as configuration or properties files
5	Cross-Site Scripting (XXS)	User's session is compromised, some action occurred related to their account that they were not aware of.	M	Attacker can steal or manipulate cookies, modify presetation of content, and compromise sensitive information.	Validate user-supplied input using positive filters to ensure it conforms to the expected format. Don't allow users to include HTML content in posts, notes, or other data that will be displayed by the application.	Throw away user input that does not fit the expected format or is HTML content.
6	Cleartext Storage of Sensitive information in Memory	Some sensitive information such as passwords has been compromised. Sensitive data is stored in plaintext.	M	Attacker can gain access to sensitive information and could compromise the system.	Avoid storing sensitive data in plaintext. Clear data after use by zeroing out the memory.	Cannot access sensitive data after it is used. Be sure it is cleared from memory.
7	Random Number Generators can be brute forced	Incorrect attempts to guess random numbers such as session keys and identifiers seen.	M	Session keys and identifiers can be brute forced alowing untrusted access to the system.	Use a trusted cryptographic random number generator.	Numbers cannot be brute forced over a specific time period.
8	Missing Encryption of Sensitive Data	Private data and cryptographic keys are exposed.	M	Cryptographic keys or private information could be exposed leading to all information and data being public.	Protect sensitive data from unnecessary exposure. Encrypt data or ensure that it is not included in the site.	Make sure all data is encrypted.
9	Use of a broken or risky Cryptographic Algorithm	Incorrect attempts to gain access to sensitive data.	M	Sensitive data could be exposed if algorithm is cracked.	Update cryptographic algorithm.	Be sure all cryptographic algorithms are up to date.
10	Directory Traversal	Files on the server have been tampered with.	M	Sensitive files can be manipulated.	Validate user-supplied input to make sure it conforms to the expected format.	Do not allow untrusted input to be evaluated or otherwise interpreted as code
11	Information Leakage	Information exposed through an error message. Log files and backup files in web-accessible directories.	L	Information about the product is leaked which could lead to a decline in the product's functionality.	Ensure only generic error messages are returned to the end user that do not reveal any additional details.	Error message returned is generic and does not reaveal any additional details to the user.
12	Untrusted Unitialization	Destination buffer overflow. Arbitrary code execution.	VL	Service can be disrupted and application could behave in unusual ways.	Limit the size of data copied from the optarg variable. Do not allow user-provide or untrusted data to control sensitive values.	Do not copy data over a certain length to be coped from the optarg variable.