Risk ID	Technical Risk	Technical Risk Indicators	Impact Rating	Impact	Mitigation	Validation Steps
1	Code can be injected into php url parameters	Improper parameters for pages including code syntax, unauthorized access to database or server	H	Leaked information, lose control of server or database, information theft, tampered / compromised server	Check php parameters for such things as <,>, php, (), so on. Cancel request or purge input	Input cleansing, if bad input detected then cancel request
2	SQL injection is possible in forms	Form submissions contain SQL syntax	H	Leaked information, access to administrator privileges	Check form submissions for SQL syntax, cleanse input or deny form altogether	Ban / suspend users, don't submit forms without cleansing first
3	Passwords for database are hard coded into code	Access to database from unauthorized users, database has been tampered with	H	Unauthorized database access, insecure data	Load passwords from remote source securely, require input	Remove the passwords from the website code.
4	Use of strcpy() in the name game	Suspicious entries to the binary, including paths, file names, executables, etc	H	Buffer overflow, undesired shell access, among other things	Prevent buffer overflow in name game to avoid nasty outcomes	Use strncpy!!!!
5	Information leakage via error message	Error message tell users what is incorrect	M	Narrow down guesses for malicious attackers, make attacking slightly easier	Don't leak any information when login fails	Print a more generic error message
6	<script> tags are not neutralized when posting user generated content	Site defacing, different images / backgrounds / noises, etc.	M /H	Site becomes unusable or obnoxious for legitimate users	Prevent scripts from being inserted into the site	Remove script tags from any and all input
7	eval() function used to evaluate url parameters	Improper parameters for pages including SQL / bash syntax (eg UNION ALL), unauthorized access to database or server	H	Unauthorized access to the database, or shell access to server, so on.	Check parameters for suspicious or not ok inputs, i.e. SQL or bash	Input checking on parameters, cancel requests if rules are violated
8	Cookie tampering	People logging in with permissions they dont actually have	H	Access to secure information or admin privileges	Keep track of user permissions and make sure their cookie permissions match	On login / page change, double check cookie permissions with DB's idea of user's permissions