Chris Piraino
Comp 116
Technical Risk Analysis

Risk ID	Technical Risk	Technical Risk Indicators	Impact Rating	Impact	Mitigation	Validation Steps
1	XSS possible	Alert messages; Redirection from webpage;	M	Insertion of arbitrary JavaScript code; Redirection to malicious website;	Escape all user input of html elements.	Test all user input fields for XSS possibility.
2	SQL Injection on Admin page	Unauthorized access to admin page; Increased volume of SQL queries;	M	Unauthorized access on website; Possible loss of data;	Sanitized all user input; Use prepared SQL query statements;	Test all user input fields for SQL injection possibility.
3	Access to arbitrary system commands .	Use of system() in URL;	H	Access to server; Possible root access; Complete access to all files;	Do not use eval() php function; Sanitize id parameters;	Ensure eval() is not used in php code.
4	Weak passwords; Brute forceable;	Increased number of incorrect logins;	H	Unauthorized access to server; Increased load on server;	Enact password policies that include numbers and special characters.	Ensure that a password must be of sufficient strength in order to be accepted.
5	Buffer Overflow in namegame	Input to namegame in binary format;	H	Execution of arbitrary shellcodes and/or other code within the process.	Remove all uses of strcpy and replace with strncpy	Ensure that strcpy is not used at any point in the code.

6	Use of Hard-coded passwords	Unauthorized access to database and/or user accounts	H	Access to root database permissions, unknown modification of database, root access to server.	Remove all plain-text/hard-coded passwords from scripts.	Ensure that no hard coded password remains.
7	Information Exposure through Error messages	Large volume of errors, attacks based on knowledge of the system.	L	Access to knowledge about the system, new attack vectors.	Change default error messages to not give away information about the system.	Check all error messages to see if any useful information is being given.