**Security Recommendations and Parameters**

**Security Recommendation 1 - Passwords:**

Passwords should be of min (6), max (12) characters.

Passwords should disallow any special characters (!, ?, @, #, $, etc.)

Passwords should be sanitized upon entry – (Remove any characters that do not fit the parameters, truncate the entry to 12 characters if longer.)

Sanitizing the passwords is important, to prevent buffer overflow attacks on our system / database. It could be possible to enter a sting of characters that may be interpreted as a command, and once it is entered into the Database, it can cause complications, loss of data, or allow a potential threat into the system.

All passwords are to be hashed using MD5 before being written to the database.

**Security Recommendation 2 – User and Employee:**

Our Front End user portal should hide, disable, obfuscate, or remove all unnecessary information, not pertinent to the end user.

(This includes things like public displaying the web server, OS, and related information to our system. End Users do not need this information, and it only makes enumeration for potential threats easier.)

Our Front End employee portal should require an additional identifier, to prevent misuse by unauthorized personnel. In addition to Username and Password, an Employee Identification Tag should be used when accessing the Employee Interface.

(This can be something as simple as a 4 or 5 digit code they have to enter upon every log in, that only employees would be knowledgeable of, and the code can change on a cyclical basis.)

**Security Recommendation 3 - Database:**

If we allow employees to make manual changes to the inventory database, we need to make sure that there is a master record database, and a backup copy database. In case of manual changes causing a loss of data, the backup needs to take the place of the main until the problem can be resolved.

Database entry should be automated, but if required to be manually entered, there needs to be a standard for data entry across any platform.

(How data needs is formatted so that no errors occur during manual entry. Ex: Standard Procedure: Employee ID / Date / Pins / Balls / Shoes / Oil. There is to be no deviation from this structure once the database has been established, and inventory is performed once.)

Passwords must be stored in Hash + Salt Format. This will be much safer for the customer’s confidential information, as it will not be stored in clear text, and if stolen, will be of no use to the attacker.

**Security Recommendation 4 – User Experience:**

Only registered users can log in, and check their previous game, high scores, and related user information.

Registered Users must play at least 1 full game of bowling, before their account is permitted to access things like scores, records, etc.

(This is remove the possible attack vector of a potential attacker creating an account to get inside the system, and see where everything is located / how it is structured. It makes sense to require a game before a new user can see scores / information pertinent to their new account.)

**Security Recommendation 5 - Lockout:**

Users should be locked out from any login attempts after no more than 5 failed attempts. Any user attempting to log in more than 5 times will face an automated error message, and the their account should be locked out, for a period of time. The user’s email on record should be alerted of the failed attempts, to notify users that their account was locked, when, and why.

**Security Recommendation 6 – Third Party Software:**

All Third-Party programs, services, or software, need to be legally acquired, enterprise versions of said software/services. These services are to be routinely updated, and tested whilst inside our environment. A secure system can become vulnerable when using third-party software or services, as what is secure on our end may not be secure on their end. These enterprise level licenses should be checked for renewal on a regular basis, so that a lapse in coverage does not occur.

When licenses are due to expire, an evaluation of that service needs to be performed, to determine if this service or software is vital to the continued operations of our systems, or is a needless cost that can be cut.

Any unauthorized use of third party materials, software, or services without an appropriate user license is a security and legal risk for our development team. Under no circumstances shall “student” or “trial” copies of software or services be integrated into our system without proper review, testing, and approval for license purchase.

**Quality Assurance Questions / Testing Procedure**

**Test 1 – Buffer Overflow Test:**

We need to perform various tests on our database and user portal, as we can succumb to a malicious attack where an attacker uses our log in forms to attempt to pass queries into our database system. We need to ensure that all forms are sanitized before being fully submitted, and checking that the sanitized entries are following the password guidelines / policies.

**Test 2 – Bad Log-in Information:**

Multiple tests should be done with bad login information, and checks should be made to ensure that there are no test accounts still available once the system goes live. Test accounts are commonly forgotten about, used for testing the log in functionality of a system, and then can serve as viable attack vectors for malicious users, gaining otherwise restricted entry into our system.

**Test 3 – Correct Username Incorrect Password:**

An error message should consistently display, informing a user that their Username OR Password are incorrect, after a failed log in. We do not want to let the attempted log in be notified if they have a genuine username or not, as this can result in a malicious actor attempting to enumerate usernames and attempt to log in or gain access with them.

(Never notify a user that their log in failed due to an incorrect password. Always obfuscate this information and never confirm if the username or password was the incorrect field.)

**Test 4 – Password Reset Options:**

There needs to be a form of security involved with resetting a password, where a user needs to provide multiple forms of identification or proof of ownership. This can include things like multiple security questions, username, user identification number, etc.

A password reset should also never be done through the system itself, but should be emailed to the user’s email on file, with further instructions or a verified link to reset the password, once authentication has been completed. This will serve as another layer of security in our system, and another layer of security for the customer and their private information.

**Test 5 – Lockout Testing:**

An account should be locked out and blocked from logging in, only if they fail to enter the correct password 5 times within a short period of time. This will prevent brute force attacks, and ensure that someone with enough time, cannot try multiple passwords and gain unauthorized access. The lockout should increase in penalty as the failed attempts increase. After 5, the lockout should last 5 minutes. After 7 attempts, the lockout should last 30 minutes, and after 10 attempts, the lockout will last an hour. Any failed attempt beyond 10 will result in a lockout for 24 hours.

**Test 6 – Database Entry – Maximum Load:**

Simple test on entering small to large amounts of data into the database to verify that it could support a standard operating load.

Multiple new accounts are created, and multiple accounts are updated, simultaneously.

**Test 7 – Database Entry – Password Hashing:**

Passwords for user accounts being stored on our database need to have SALT added, and the resulting string of characters will be hashed using MD5. Is our database capable of removing the salt and comparing the passwords for validation, in a quick and timely manner? Do we need to continue to change our SALT added, or will this hamper the performance?

**Test 8 – User - Payment:**

When collecting forms of electronic payments, we need to be sure the automated pricing for the bowling games is calculated correctly. This includes number of players, number of games played, and number of shoe rentals. We are to test our calculations and assure that every possible number of players option is billed correctly (1-5).

(Ex: 1 game = $3.00 / 4 bowlers = $12 – Every time it is performed.)