Web-Based Evaluations

Security Manual

Class CPSC 488 Section 2

Group Number: 2

Revised: 4/22/2023

Originally Created: 4/25/2022

Tanuj Rane txr1029@sru.edu

Dalton Stenzel drs1030@sru.edu

Logan Racer lsr1006@sru.edu

Anthony Cinicola ajc1033@sru.edu

Duncan Lawrence dal1017@sru.edu

Revised By:

Chad Green cwg1003@sru.edu

# Contents

[Contents 2](#_1o161ow89tho)

[Introduction 3](#_gjdgxs)

[Login 3](#_30j0zll)

[Activity Logging 3](#_bsibnco6tbwz)

[Passwords 4](#_1fob9te)

[Role Division 4](#_3znysh7)

[User Creation Privileges 5](#_xipv9pdil9sw)

[Page Access 5](#_imutwmm7wpzg)

[Exception Catching 5](#_2et92p0)

[Flaws 6](#_tyjcwt)

# Introduction

The security manual for this specific software is designed to cover the key points of how the system either is or should be protected from situations such as errors and hacking. Each section below will cover a particular topic of security threats pertaining to the software and how they’re resolved. The very last section will cover security flaws in our system that need to be taken into consideration for the next set of hands that are determined to pick up where we left off.

# Login

First topic to be discussed is the action of logging in. When a new user is created, a temporary password is created on their behalf from a user administrator. Once the user is informed of their temporary password, they will be prompted to reset their password upon logging in for the very first time. In order to prevent mistakes, the “First Login Screen” will provide a box to reenter the old password as well as two password boxes that prompt the user to enter the new password. By default, the characters being typed into the boxes are hidden in order to prevent onlookers from seeing what the user is entering. The two boxes relating to the new password must contain exactly matching entries in order for a user to reset or set their password..

This “First Time Login” section is integrated into the system as a means to protect newly created users from having a password created by someone else as well as to allow the users to set their own more secure passwords. In the case of a “user upload file”, a file used to create new users upon being uploaded, being leaked, the passwords are partially protected as those files exclusively contain hashed versions of the temporary passwords used for users. As long as the file doesn’t use the same hashed password for all the users, then the users will be protected from having their accounts stolen in such a way.

Again as stated before, there has now been a new implementation that requires users to enter their old password when they enter the “First Time Login” screen. By requiring the users to enter their old password we can verify that the user is who they say they are and thus update the old password securely. The old password is entered in the first text box, followed by the new password in the next two boxes.

# Activity Logging

The ability to obtain log files relating to information done from within the system is accessible through the “ADMIN” role from any page in the navigation bar. The logging capabilities of the program include, but aren’t limited to: users who log in, user account changes, rows not added from a user add file, group creations, group edits, when an evaluation is started/attempted/finished, uploading an evaluation file with or without warn/issues, etc.

# Passwords

Continuing the topic of login details, passwords are protected via the use of the BCrypt hashing function. A hashing function can be described as a function used to produce coded values used as an index from an arbitrary value. The index value serves to be the value checked when a password is entered. Upon entering a password to login, the password is checked with the password stored for the user email that is being used to login in. If the user-entered password correctly points to the same index where the hashed password is, then it is a match and the user is logged in, otherwise a “login failed”-esque message displays. When editing, resetting, or manually adding a user, any password submitted must be five or more characters, where spaces aren’t allowed nor counted. Failure to make the password at least five characters long with no spaces will result in feedback mentioning the failure to submit the information. The only exception to adding small passwords is when a user is added via the “upload users” ability where there isn’t currently a way to check the number of characters provided as the password in the file is a hashed variant of the password so retrieving the original password’s length is impractical as of now. This measure is preferred as a smaller temporary password that is protected via a hash function is a lot safer than having a user’s raw temporary password located somewhere in an unprotected format.

# Role Division

When it comes to security, there should be some form of privilege given to different users in order to split up who has access to what in the program. This has been done by creating some different roles that have access to different pages. The “SuperSuperUser” is the head admin of the system. This user has access to the manage companies page as well as to the manage users page. However, the SuperSuperUser only has access to edit the details of other admins for the sub companies that are utilizing the service. The SuperSuperUser is able to add companies via a company upload file as well as activate and deactivate current companies entered in the system. The SuperSuperUser also has the privilege to add new admins to existing companies and edit existing companies.

Another role is the “SuperUser” or “ADMIN” role. An admin is responsible for managing roles and users related to their respective company. Roles can be added via an upload roles file and then edited on the manage roles page. User info can be edited and users can be added manually or removed manually via the manage users page. This role is different from the “EVAL\_ADMIN” role which gives a user access to the management of evaluation forms and groups. An EVAL\_ADMIN can upload evaluation forms via the Evaluation Forms page and can upload groups via the Groups page. Groups are uploaded using an upload groups file or can be created manually. In addition, groups can be edited, viewed, archived, and deleted from this page by the EVAL\_ADMIN. The last 2 roles include USER and EVALUATOR. The USER only has access to their account information and their evaluations. EVALUATOR users can access evaluation groups and complete evaluations relating to the users.

# User Creation Privileges

A recent addition to the program is the addition of some privilege controls SuperSuperUsers can now only create new admin users and only view the admin users that are in the system. Previously the SuperSuperUser could view all of the users and all of the admins but they can now only view admins. They can also only create admins and the creation of any other user is not allowed for the SuperSuperUser. There has also been a change to what users ADMINS have access to. they now only have access to the users in their own company as well as their own account data. they can no longer see the SUperSuerUser and they can no longer delete their own account. They thus only have access to the users in their respective company and their own details. This enhancement improves the security of the system as the SuperSuperUser can now no longer access or delete the users in a respective company and the ADMINS can no longer access the users with higher privileges such as the SuperSuperUser or the other admins from other companies.

# Page Access

This program involves the need to upload forms and files in a specific order so that everything is populated correctly. In the past an ADMIN could upload whatever files they wanted and were presented with errors due to the order of events being incorrect. This has been changed by implementing page buttons that cannot be accessed until each step is completed in order. Companies must be uploaded in order for the manage users page to be accessed by any of the admins or the SuperSuperUser themselves. Once this is completed the page can then be accessed by the SuperSuperUser however, the ADMIN will still not be able to access the manage users page until they themselves upload roles. Once roles are uploaded the manage users page can then be utilized. During this stage of upload the groups page is also inaccessible. However, once the users are successfully uploaded, the groups page can then be accessed in order to create groups using the users that are now populated in the system. By making this kind of change to the system the program is being protected and the user is protected from making errors during setup.

# Exception Catching

As with any piece of software, errors may and likely will occur. Having such a problem can lead to unintentional backdoors, data leaking, or something of the sort. Regardless, taking action to prevent errors, faults, and failures are necessary to avoid costly and irreparable damage. Exception catching is a tool used to help prevent crashes for a system. While it hasn’t been implemented for every single method in each class there are some instances of its use in methods that require computation. One such example is in the pageCalc() method in the AdminMethodsService class in order to catch any weird errors that might occur if something like an index is amiss.

# Flaws

Time constraints and limited knowledge has led to some shortcuts and unfinished business left unattended. Notable design flaws with our program include a lack of an “ADMIN-LOGS” role in order to view and control the logging being done, SQL injection tests, failure to restrict urls and strong password requirements.

To expand on some of these flaws, it would be beneficial to have a ADMIN\_LOG as it would allow only one separate user to review log activity and see what activity is being done. The current state of the program allows the admins of any kind to access this data when they really do not need to have it. that is more so a security feature that should only be reviewed by a security user.

SQL injection could be a problem with this program. There are a lot of instances where people are entering data and could enter malicious text such as an SQL injection that would allow a malicious user access to the database and possibly a higher privilege.

There is also a failure to restrict urls in this program. If a normal user gets access to a URL related to a higher privileged page the user can simply enter that URL and have access to that page. There should be some form of access control for the urls themselves that only allows certain users access to those pages. That currently is not the case but could be implemented in the future.

Lastly, there currently is no requirement for strong passwords. The current passwords only need to be longer than 5 characters. Most passwords by today's standards need to be at least 8 characters in length and include an uppercase letter, lowercase letter, number, and special character with no repeating characters.