Thomas Simmons, Ricky Matherly, Jalen Keller, Noah Caldwell

CSCI 4628 Group Project Draft

**Abstract:**

We intend to develop a secure United States Presidential voting system. To accomplish this we will develop a Java application that provides a GUI for the user to choose from a list of candidates as well as write in a candidate that is not listed. Upon submitting, the user’s vote will be kept track in a MySQL database. This application will emphasize strong defensive programming to ensure as little tampering as possible. We intend to develop in a windows environment. ChatGPT will be used to expedite the development with the Professor’s permission.

**Thomas Simmons & Noah Caldwell - User Input Validation and Security**

Whenever an application uses user input, it opens itself up to security vulnerabilities. Hackers will attempt to use surfaces of user input to their advantage. Without user validation, hackers can inject malicious code into the application. Input validation is also important for non-malicious users. Common users may not give the input developers are expecting. This can lead to unintended consequences and potentially break the program. Good input validation should account for every possible scenario, especially when it could lead to a security breach. Input validation should be more important than the user experience.

The purpose of this project is not to prove that we can make a voting system, but that we can make a secure voting system. This means input validation plays an important role in ensuring the security of our application. We will use input validation to prevent users from manipulating our voting system. User experience will not be sacrificed in exchange for any potential security threat. We will be checking that the input is of the required type, of the expected length, and of the necessary authentication. If users are allowed to submit multiple votes or injected code, the security of the election would be compromised. Key techniques we will be implementing include sanitization, buffer overflow prevention, and relevant error messaging.

**Ricky Matherly - Java GUI**

Java GUI is a crucial component when creating the presidential voting system, and the central part is to display the entire election. It is summarized into four major parts. Panels, Frames, Labels, and Buttons:

1. Panels are the main background for GUI, and it is displayed with a white background, making it clearer to see. Different sizes, ranging from the X and Y axis, can change the Panel's position. The vital point for the Panel in the voting system is to display the whole layout for the entire election.
2. Frames are similar to Panels; however, it is used for Labels and Buttons to be displayed upon them. Frames are sometimes an alternative to Panels; they are both combined to show a complete GUI. The utter point of Frames is to display the election while using buttons and labels.
3. Labels are used for background text format. For this instance, it is to display the names, the title, and other details.
4. Buttons are used to initiate a function, most occasionally an action listener. The action listener, on most occasions, displays a message telling the user that the process has worked.

**Jalen Keller - Database**

Databases are the biggest part of storing all kinds of information. It’s used by almost every company in the world to store things like company information or even customer information. Databases have to be secured to the best of your ability because all information can be used by somebody and they will take advantage if it’s possible to. There will be so much potential money being lost if you don’t make the smart investment of paying for a secure database.

The database will need to be secure to prevent unauthorized reads or writes. The database will be where we store sensitive personal information as well as the tally of votes for each candidate. The candidate a user votes for will stay anonymous in accordance with the law. We will not keep track of this information because of any potential data breach. The database will be hosted using XAMPP for this project along with an Apache web server on our local machines. All of the information will be secured tightly within a MySQL database.

The connection to the database will need to be protected to prevent any sniffing, man-in-the-middle attacks, or any other malicious intrusions. We will focus on heavy backend security to ensure the security of our database queries. The information will be separated in different columns like: Candidate names and votes. The database should be very accessible to the rest of the code.

**References:**

Seacord, Robert C. Secure Coding in C and C++. 2nd ed., Addison-Wesley Professional, 2013.

OWASP Foundation. "OWASP Secure Coding Practices Quick Reference Guide." OWASP,

2021, https://owasp.org/Top10/A7\_2021-Cross-Site\_Scripting\_(XSS).html#references.