Mackenzie Bonneville

Table 6.8

	Yes	**No**	**N/A**	**Comments/Evidence/Rationale**
Authentication				
1. Does each Web request validate authentication?			x	
2. Are credentials presented securely (i.e. using SSL, not the GET method)?			x	
3. Are passwords stored in an encrypted or hashed format?	x			When passwords are validated, it checks the hashes of the password.
4. Is password complexity enforced, including minimum length, non-guessable words, special characters, numbers?	x			Password length is checked and must be over 8 characters in length. There are not more complexity checks enforced though.
5. Do credentials expire after a period of time?			x	
6. Are standards used for authentication and identity management (i.e. SAML, WS-Security, LDAP, NTLM, Kerberos?)			x	
7. Are user accounts locked after a certain number of failed attempts?			x	Accounts are not locked however after 5 failed passwords the user is kicked from the system.
Authorization				
8. Are permissions defined to create fine-grained user access?			x	
9. Are permissions defined to create fine-grained administrator access?			x	

10. Are permissions enforced consistently in the application?			x	
11. Can permissions be grouped or organized to user roles for simplified access management?			x	
12. Are roles and permissions consistent with standards or other applications in the enterprise?			x	
Data Validation				
13. Are all user inputs validated?	x			Input length is verified.
14. Does validation check data length?	x			Data is read into heap memory so the odds of overflow are slim.
15. Does validation filter or escape special characters?		x		Special characters have no affect on the input.
16. Does validation of web input remove tags before displaying it back to the user?			x	
17. Does the application validate the data type of user input before operating on it?			x	
18. Is XML received from outside of the application validated?			x	
19. Is the integrity of files sent and received by the application validated?	x			Yes, the server checks the hash of the original message before executing any type of code onto the information received.
Session Management				
20. Is session data excluded from the URL using the GET method?			x	

21. Does the data in the browser cookie contain only the session ID and exclude other session information?			x	
22. Are session IDs hashed to prevent attackers from guessing valid session IDs?			x	
23. Are session IDs guaranteed to be unique?			x	
24. Are sessions validated on each page request?			x	
25. Do sessions expire after a period of inactivity?		x		Connections stay open until exited manually or when the code has completed its run.
26. Are expired sessions deleted on the server?			x	
Logging				
27. Are security-related events logged consistently?		x		There is no specific log file. Security events are printed to console.
28. Is sensitive information, such as passwords, kept from logs?	x			There are no logs
29. Are security events stored in a secure location and not mixed with common application logging?		x		There is not a log file
30. Are events logged in a format and location that is compatible with security monitoring/event correlation software?		x		There is not a log file
Error handling				
31. Are exception	x			Exit will happen upon error, skipping the

handling mechanisms used consistently?				remaining code, and displaying the error code to the console.
32. Does the application fail securely? If so, how?	x			See above answer.
33. Are open transactions process appropriately if an error is encountered during processing?			x	
34. Are error messages displayed to the users informative without revealing information about system internals or other sensitive data?	x			Error messages do not contain any sensitive data.
35. For function-based error handling, are return values of functions tested?	x			Yes, for every function call, the return value is tested before the code continues.
36. For exception-based error handling, are specific exceptions caught rather than broad exception handlers (i.e., throwable in Java)?			x	
37. Are exceptions that are caught managed and logged (i.e no empty catch{} blocks)?			x	
Cryptography				
38. What is the sensitivity of the data being processed by the application?				The data in this program is simple and not sensitive.
39. Is encryption required for the data? If so, in transit, at rest, or both?		x		Although not required, it is used in the transfer of the original password.
40. Does the application comply with your organization’s standards			x	

regarding encryption?				
41. Are standard, accepted encryption protocols being used rather than home-grown algorithms?	x			OpenSSL’s AES library is being used with a 256bit key.
42. Are passwords encrypted in transit and at rest?	x			They are encrypted in transit.
43. Are keys used with encryption protocols managed securely in the application?		x		They are stored in plaintext in the code.
Performance				
44. Is this application thread-safe?	x			Program uses pthread to handle different variables at the same time without interference.
45. Are variables encapsulated to limit their scope and prevent sharing between processes?		x		Variables are used directly or globally defined.
46. Are efficient algorithms used?	x			
47. Are database transactions clearly defined and not subject to deadlocks?			x	No database was used for this project.
48. Are database tables indexed properly?			x	
49. Are file handles and connections to external systems explicitly closed?	x			Files are closed as soon as use of them is complete.
50. Are all variables that are initialized actually used?	x			All variables are used within the code.