(DB) Address Book Appliance Security Test Report

Prepared by: Pelumi Akande | Ross Barber | Carina Martinez-Lopez | Zoe Medina | Randall Seymore

For issues with embedded spreadsheets or links, please use the Google Doc
https://docs.google.com/document/d/1CMYnd9OsctJFz73PRUU20QHV_oiSl6xOD64_bgYvMHA/edit?usp=sharing

Table of Contents

Security Requirements

Table of identified security requirements used to develop Test Scenarios and Test Cases.

Requirement ID	Requirement
SR-1	**Login credential complexity** - The system shall require that the login credentials are non-trivial and unpredictable, to mitigate password guessing, dictionary and other brute-force attacks.
SR-2	**Account lockout** - After three consecutive incorrect login attempts, the system shall generate an alarm and also lock-out the account for 1 minute, to mitigate dictionary and other brute-force attacks.
SR-3	**Authentication** – The system shall require users to authenticate themselves before granting access to system resources.
SR-4	**Access rights** - The system shall only allow access to system resources to authenticated users with the proper authorization
SR-5	**Sensitive Information Protection** – The system shall protect sensitive information from unauthorized disclosure while it is stored or in transit

Test Scenario Summary

Table of Test Scenarios mapped to identified security requirements, level of importance, and number of test cases conducted within the scenario.

Test Scenario ID	Requirement	Test Scenario Description	Importance	# of Test Cases
TS-1	SR-1	Validate if passwords can be guessed through brute-force attacks given the current password requirements	Medium	4
TS-2	SR-2	Validate the system generates an alarm that locks out accounts for 1 minute following 3 consecutive failed login attempts.	Medium	1
TS-3	SR-3	Enforce mechanisms requiring users to authenticate before granting access to system resources on the ABA	High	5
TS-4	SR-4	Validate that users cannot bypass the system and access admin-level privileges.	High	5
TS-5	SR-5	To verify that the system effectively safeguards sensitive information against unauthorized disclosure, both while it is stored within the system and during data transmission.	High	4

Test Scenario 1

Test cases for Test Scenario 1 (TS-1) apply to Security Requirement 1 (SR-1) for Login credential complexity which requires the system to require that the login credentials are non-trivial and unpredictable to mitigate password guessing, dictionary and other brute-force attacks.

TS-1 Test Case 1

Test Case ID: **TC-SR1-01**	
Test Priority: Medium	Test Designed By: Zoe Medina
Module Name: ABA Password Complexity	Test Designed Date: 09/29/23
Test Title: Common Passwords	Test Executed By: Zoe Medina
Description: Testing the allowance of top common passwords during account creation	Test Executed Date: 10/1/23
Preconditions: n/a	
Dependencies: n/a	

Execution

Step	**Test Steps**	**Test Data**	**Notes**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
1	Add a new user through the admin account	Top 5 Common Passwords of 2023: 123456 123456789 qwerty password 12345		Program should return completion code "password is too easy to guess" and reject the inputted password	Program returned completion code "password is too easy to guess" and rejected the inputted password	**Pass**
2	Log in as the new user for the first time					
3	Set password to one of the common passwords					
4	Repeat step 2-3 for each common password					

TS-1 Test Case 2

Test Case ID: **TC-SR1-02**	
Test Priority: Medium	Test Designed By: Zoe Medina
Module Name: ABA Password Complexity	Test Designed Date: 09/29/23
Test Title: Brute-Force Attack	Test Executed By: Zoe Medina
Description: Testing the resistance of passwords against brute-force attack given current password requirements	Test Executed Date: 10/1/23
Preconditions: User account has been created and a password has been set up meeting the minimum security requirements	
Dependencies: n/a	

Execution

Step	Test Steps	Test Data	Notes	Expected Result	Actual Result	Status (Pass/Fail)
1	Create a wordlist of all combinations of numbers, lowercase, and uppercase with lengths of 1-8	userID = zoe	Python script modified to have a static userID	Brute-force script should not successfully log in and print password	Brute-force script successfully logged into account and printed the found password	**Fail**
2	Set static userID to "zoe"	password = q				
3	Run brute-force script against ABA login function					

TS-1 Test Case 3

Test Case ID: **TC-SR1-03**	
Test Priority: Medium	Test Designed By: Zoe Medina
Module Name: ABA Password Complexity	Test Designed Date: 09/29/23
Test Title: Common Passwords (Admin)	Test Executed By: Zoe Medina
Description: Testing the allowance of top common passwords during admin account creation	Test Executed Date: 10/1/23
Preconditions: Admin account has not been set up yet	
Dependencies: n/a	

Execution

Step	Test Steps	Test Data	Notes	Expected Result	Actual Result	Status (Pass/Fail)
1	Launch ABA for the first time	Top 5 Common Passwords of 2023: 123456 123456789 qwerty password 12345		Program should return completion code "password is too easy to guess" and reject the inputted password	Program returned completion code "password is too easy to guess" and rejected the inputted password	**Pass**
2	Log in as the admin user					
3	Set password to one of the common passwords					
4	Repeat step 2-3 for each common password					

TS-1 Test Case 4

Test Case ID: **TC-SR1-04**	
Test Priority: Medium	Test Designed By: Zoe Medina
Module Name: ABA Password Complexity	Test Designed Date: 09/29/23
Test Title: Brute-Force Attack (Admin)	Test Executed By: Zoe Medina
Description: Testing the resistance of password against brute-force attack given current admin password requirements	Test Executed Date: 10/1/23
Preconditions: Admin account has been set up with a password meeting the minimum password requirements	
Dependencies: n/a	

Execution

Step	**Test Steps**	**Test Data**	**Notes**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
1	Create a wordlist of all combinations of numbers, lowercase, and uppercase with lengths of 1-8	userID = admin	Python script modified to have a static userID	Brute-force script should not successfully log in and print password	Brute-force script successfully logged into account and printed the found password	**Fail**
2	Set static userID to "admin"	password = 8				
3	Run brute-force script against ABA login function					

Test Scenario 2

Test cases for Test Scenario 2 (TS-2) apply to Security Requirement 2 (SR-2) for Account lockouts, requiring the system to generate an alarm and lock-out the account for 1 minute after three consecutive incorrect login attempts, to mitigate dictionary and other brute-force attacks.

TS-2 Test Case 1

Test Case ID: **TC-SR2-01**		
Test Priority: Medium		Test Designed By: Randy Seymore
Module Name: Account Lockout		Test Designed Date: 10/01/2023
Test Title: Account Locks Out		Test Executed By: Randy Seymore
Description: Testing enforcement of account lockout after three consecutive unsuccessful login attempts		Test Executed Date: 10/01/2023
[redacted]		
Preconditions: admin account has been created and login credentials have been set, adhering to password complexity requirements discussed in TS-1.		
Dependencies: n/a		

Execution

Step	Test Steps	Test Data	Notes	Expected Result	Actual Result	Status (Pass/Fail)
1	Launch ABA	User Login field displays		Alert message will display "Account locked due to 3 failed login attempts"	No limits are applied to failed login attempts. Therefore no alarm is generated, resulting in a failure.	**Fail**
2	Attempt Login three consecutive times using incorrect password each time	userID = admin Password = 3 consecutive bad passwords				
3	Verify alarm is generated	Alarm = “Account locked out due to three consecutive failed login attempts”				
4	Start a timer when the alarm is generated. Verify 1 minute threshold.	Account lockout duration is 1 minute				
5	Attempt to input data into the login field during lockout	Login field locks, not allowing any input to be entered				

6	After 1 minute, attempt additional login	Login field unlocks allowing for login attempts (3) to resume				🟥

Test Scenario 3

Test cases for Test Scenario 3 (TS-3) apply to Security Requirement 3 (SR-3) for Authentication, requiring the system to require users to authenticate themselves before granting access to system resources.

TS-3 Test Case 1

Test Case ID: **TC-SR3-01**	
Test Priority: Low	Test Designed By: Ross Barber
Module Name: Login (LIN)	Test Designed Date: 10/1/2023
Test Title: Valid User Login	Test Executed By: Ross Barber
Description: Verify that the system authenticates valid users and grants them access to system resources.	Test Executed Date: 10/2/2023
Preconditions: ABA login prompt is accessible to a user, and the user has valid credentials	
Dependencies: User database and authentication service are operational	

Execution

Step	Test Steps	Test Data	Notes	Expected Result	Actual Result	Status (Pass/Fail)
1	Reach the ABA CLI login prompt			The user should be authenticated and granted access to the system resources.	Specified user was authenticated to the system with the valid credentials	**Pass**
2	Complete the prompts to input a valid username and password	- Username: ValidUser - Password: ValidPassword1	Ensure the user account is active and the password is correct.			
3	Submit authentication request					

TS-3 Test Case 2

Test Case ID: **TC-SR3-02**	
Test Priority: Medium	Test Designed By: Ross Barber
Module Name: Login (LIN)	Test Designed Date: 10/1/2023
Test Title: Invalid Username/Password Login	Test Executed By: Ross Barber
Description: Ensure that the system denies access for login attempts with an invalid username or password	Test Executed Date: 10/2/2023
Preconditions: ABA login prompt is accessible to a user	
Dependencies: User database and authentication service are operational	

Execution

Step	Test Steps	Test Data	Notes	Expected Result	Actual Result	Status (Pass/Fail)
1	Reach the ABA CLI login prompt			The system should deny access and display the appropriate error message for either invalid parameter passed	"Invalid credentials" error received when providing nonexistent user "Invalid credentials" error received when providing invalid password after inputting a valid user	**Pass**
2	Complete the prompts to input an invalid username	Username: InvalidUser	Ensure the username **does not** exist in the system. Inputting a nonexistent username should present an error prior to password prompt			
3	Submit authentication request					
4	Complete the prompts to input a valid username with an invalid password	- Username: ValidUser - Password: InvalidPassword1	Ensure the username **does** exist in the system. Inputting a legitimate username with an invalid password should present an error.			
5	Submit authentication request					

TS-3 Test Case 3

Test Case ID: **TC-SR3-03**	
Test Priority: Medium	Test Designed By: Ross Barber
Module Name: Login (LIN)	Test Designed Date: 10/1/2023
Test Title: Empty Username/Password Login	Test Executed By:
Description: Ensure that the system denies access when either username or password field is left blank during a login attempt	Test Executed Date:
[redacted]	
Preconditions: ABA login prompt is accessible to a user	
Dependencies: User database and authentication service are operational	

Execution

Step	**Test Steps**	**Test Data**	**Notes**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
1	Reach the ABA CLI login prompt			The system should deny access and display the appropriate error message for either invalid parameter passed	"Missing parameter" error received when providing a blank username "Invalid credentials" error received when providing blank password after inputting a valid user	**Pass**
2	Input no value when prompted for username	Username:	Ensure no data is input when prompted			
3	Submit authentication request					
4	Complete the prompts to input a valid username, inputting no values when prompted for a password	- Username: ValidUser - Password:	Ensure no data is input when prompted for password only			
5	Submit authentication request					

TS-3 Test Case 4

Test Case ID: **TC-SR3-04**	
Test Priority: Medium	Test Designed By: Ross Barber
Module Name: Change Password (CHP)	Test Designed Date: 10/1/2023
Test Title: Validate Password Change Function	Test Executed By:
Description: Verify only authenticated users can change passwords	Test Executed Date:
Preconditions: User is initially not authenticated to the ABA and has been granted access to change password	
Dependencies: User database, authentication service, and password change function are operational	

Execution

Step	Test Steps	Test Data	Notes	Expected Result	Actual Result	Status (Pass/Fail)
1	Reach the ABA CLI			The password should be successfully changed only after the user has authenticated with valid credentials.	"No active login session" error received when attempting to call the CHP function before authenticating to the ABA CHP called successfully after authenticating with Test Data credentials	**Pass**
2	Call Change Password (CHP) function					
3	Complete the prompts to input a valid username and password	- Username: ValidUser - Password: ValidPassword1				
4	Submit authentication request					
5	Call Change Password (CHP) function					

TS-3 Test Case 5 **(Extra Credit)**

Test Case ID: **TC-SR3-05**	
Test Priority: High	Test Designed By: Zoe Medina
Module Name: Login (LIN)	Test Designed Date: 10/2/2023
Test Title: Unauthorized access	Test Executed By: Zoe Medina
Description: Gain access to other user's account using a different account	Test Executed Date: 10/2/2023
Preconditions: Two accounts exist within the system	
Dependencies: n/a	

Execution

Step	Test Steps	Test Data	Notes	Expected Result	Actual Result	Status (Pass/Fail)
1	Enter "LIN test admin"	user=admin password=hi		WAI command should show that test account is logged into or should reject the entered username	WAI command shows that admin account is logged into	**Fail**
2	Enter the password for test account	user=test password=bye				
3	Enter "WAI" command					

Test Scenario 4

Test cases for Test Scenario 4 (TS-4) apply to Security Requirement 4 (SR-4) for Access rights, requiring the system to only allow system resource access to authenticated users with the proper authorization.

TS-4 Test Case 1

Test Case ID: **TC-SR4-01**	
Test Priority: High	Test Designed By: Carina Martinez-Lopez
Module Name: Access Rights	Test Designed Date: 09/29/23
Test Title: Admin Login	Test Executed By: Carina Martinez-Lopez
Description: Testing to see if the admin user can has access to admin level rights.	Test Executed Date: 10/02/23
Preconditions: n/a	
Dependencies: n/a	

Execution

Step	Test Steps	Test Data	Notes	Expected Result	Actual Result	Status (Pass/Fail)
1	Open OS and login as the administrator.	OS Home Page		The system should allow the administrator to login using their admin username and password. They, the admin should then be able to access admin level rights to include command line admin commands, changing roles and more.	Results were successful and turned out as we expected.	**Pass**
2	Use correct username and password to login.	Admin Username: admin Admin Password: ******				
3	Login will be successful.	Successful login.				
4	Admin user will attempt to access admin privileges such as changing their user rights.	Successful admin right privileges.				

TS-4 Test Case 2

Test Case ID: **TC-SR4-02**	
Test Priority: High	Test Designed By: Carina Martinez-Lopez
Module Name: Access Rights	Test Designed Date: 09/29/23
Test Title: Admin Login Attempt with User Username	Test Executed By: Carina Martinez-Lopez
Description: Testing to see if the admin user can has access to admin level rights logging in with an user username and admin password.	Test Executed Date: 10/02/23
[redacted]	[redacted]
Preconditions: n/a	
Dependencies: n/a	

Execution

Step	**Test Steps**	**Test Data**	**Notes**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
1	Open OS and login in..	OS Home Page		The system should not allow the administrator to login using a user username and admin password. They should not be able to access admin level rights to include command line admin commands, changing roles and more.	Results were successful and turned out as we expected.	**Pass**
2	Use in-correct username and correct admin password to login.	Username: user Password: ******				
3	Login will be unsuccessful.	Unsuccessful login.				
4	User will not be able to access admin rights.	Unsuccessful admin right privileges.				

TS-4 Test Case 3

Test Case ID: **TC-SR4-03**	
Test Priority: High	Test Designed By: Carina Martinez-Lopez
Module Name: Access Rights	Test Designed Date: 09/29/23
Test Title: Login Using Admin Username and User Password	Test Executed By: Carina Martinez-Lopez
Description: Testing to see if users can log in using the admin username and user password.	Test Executed Date: 10/02/23
[redacted]	
Preconditions: n/a	
Dependencies: n/a	

Execution

Step	Test Steps	Test Data	Notes	Expected Result	Actual Result	Status (Pass/Fail)
1	Open OS and attempt to login.	OS Home Page		The system should not allow the administrator to login using an admin username and user password. They should not be able to access admin-level rights to include command line admin commands, changing roles and more.	Results were successful and turned out as we expected.	**Pass**
2	Use admin username and user password to login.	Username: admin Password: 12345				
3	Login will be unsuccessful.	Unsuccessful login.				
4	User will not have access to admin privileges.	Unsuccessful admin right privileges.				

TS-4 Test Case 4

Test Case ID: **TC-SR4-04**	
Test Priority: High	Test Designed By: Carina Martinez-Lopez
Module Name: Access Rights	Test Designed Date: 09/29/23
Test Title: User Login Attempt to Access Admin Command Line	Test Executed By: Carina Martinez-Lopez
Description: Testing to see if the user can has access to admin level rights form the command line.	Test Executed Date: 10/02/23
Preconditions: n/a	
Dependencies: n/a	

Execution

Step	Test Steps	Test Data	Notes	Expected Result	Actual Result	Status (Pass/Fail)
1	Open OS and login as the an user.	OS Home Page		The system should not allow the user to admin-level rights through the command line when logged in as an user.	Results were successful and turned out as we expected.	**Pass**
2	Use correct user username and password to login.	Username: user Password: 12345				
3	Login will be successful.	Successful login.				
4	User will attempt to access admin privileges through the command line.	Unsuccessful admin right privileges.				

TS-4 Test Case 5

Test Case ID: **TC-SR4-05**	
Test Priority: High	Test Designed By: Carina Martinez-Lopez
Module Name: Access Rights	Test Designed Date: 09/29/23
Test Title: User Login Attempt to Give Themselves Admin Rights	Test Executed By: Carina Martinez-Lopez
Description: Testing to see if the user can give themselves access to admin level rights.	Test Executed Date: 10/02/23
Preconditions: n/a	
Dependencies: n/a	

Execution

Step	Test Steps	Test Data	Notes	Expected Result	Actual Result	Status (Pass/Fail)
1	Open OS and login as the user.	OS Home Page		The system should not allow the user to give themselves admin-level rights as a user.	Results were successful and turned out as we expected.	**Pass**
2	Use the correct user username and user password to login.	Username: user Password: 12345				
3	Login will be successful.	Successful login.				
4	User will attempt to give themselves admin privileges.	Unsuccessful admin right privileges.				

Test Scenario 5

Test cases for Test Scenario 5 (TS-5) apply to Security Requirement 5 (SR-5) for Sensitive Information Protection, requiring the system to protect sensitive information from unauthorized disclosure while it is stored or in transit.

TS-5 Test Case 1

Test Case ID: **TC-SR5-01**	
Test Priority: Low	Test Designed By: Pelumi
Module Name: LIN / REA	Test Designed Date: 10/01/2023
Test Title: Authorized Record Access	Test Executed By: Ross Barber
Description: Permit authenticated users to read owned records	Test Executed Date: 10/02/2023
Preconditions: User with valid authorization	
Dependencies: n/a	

Execution

Step	Test Steps	Test Data	Notes	Expected Result	Actual Result	Status (Pass/Fail)
1	Reach the ABA CLI login prompt			ABA Records accessible to authorized users	Following successful LIN, the specified user was permitted to call REA	**Pass**
2	Complete the prompts to input a valid username and password	- Username: ValidUser - Password: ValidPassword1				
3	Submit authentication request and issue the ADR <recordID> command	RecordID = 100				
4	Issue the REA command to list all records for the logged in User					

TS-5 Test Case 2

Test Case ID: **TC-SR5-02**	
Test Priority: High	Test Designed By: Pelumi
Module Name: LIN / REA	Test Designed Date: 10/01/2023
Test Title: Unauthorized Record Access	Test Executed By: Ross Barber
Description: Verify that unauthorized users cannot access sensitive information stored within the system	Test Executed Date: 10/02/2023
[redacted]	
Preconditions: User without proper authorization.	
Dependencies: n/a	
[redacted]	

Execution

Step	Test Steps	Test Data	Notes	Expected Result	Actual Result	Status (Pass/Fail)
1	Reach the ABA CLI login prompt			ABA Records are not accessible to unauthenticated users	"No active login session" error received when calling the REA function as an unauthenticated user	**Pass**
2	Issue the REA command to list all records for the logged in User					
[redacted]						

TS-5 Test Case 3

Test Case ID: **TC-SR5-03**	
Test Priority: Low	Test Designed By: Pelumi
Module Name: N/A	Test Designed Date: 10/01/2023
Test Title: Data Encryption in Transit	Test Executed By: Ross Barber
Description: Verify that data transmission over secure protocols	Test Executed Date: 10/02/2023

encrypts sensitive information.	
[redacted]	
Preconditions: The system can transmit sensitive information to another system or location. Test User on source ABA host has an existing database and permissions to export it	
Dependencies: Linux system with pcap to run the ABA; A system or component to send or receive sensitive information	
[redacted]	

Execution

Step	Test Steps	Test Data	Notes	Expected Result	Actual Result	Status (Pass/Fail)
1	Reach the ABA CLI login prompt			Packets captured will maintain integrity of the database contents by remaining encrypted in the event they are viewed on the wire	Data payload could not be read in packets intercepted with pcap	**Pass**
2	Complete the prompts to input a valid username and password	- Username: ValidUser - Password: ValidPassword1				
3	Using a separate shell session, initiate a packet capture on the Linux server hosting the ABA application					
4	Export the Records database					
5	SCP the database file to the destination host		Functionality not in place on current application release to transmit files to remote host. SCP used to emulate transmission of files over a secure protocol			

TS-5 Test Case 4

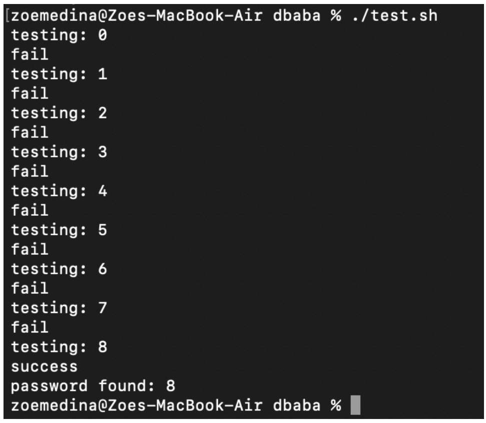
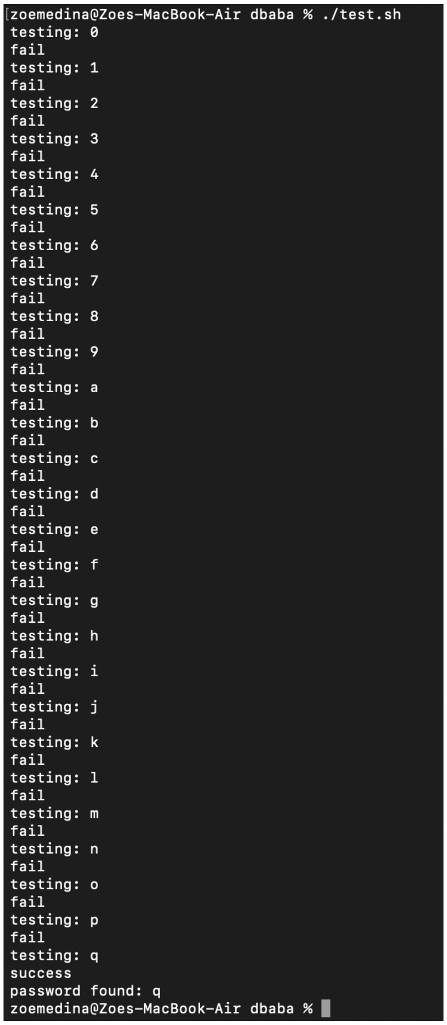
Test Case ID: **TC-SR5-04**	
Test Priority: High	Test Designed By: Pelumi
Module Name: N/A	Test Designed Date: 10/01/2023
Test Title: Data Encryption at Rest	Test Executed By: Ross Barber
Description: Verify that sensitive information stored on the ABA is protected with proper encryption	Test Executed Date: 10/02/2023
Preconditions: ABA host has an existing database populated with data. Sqlite installed on ABA host	
Dependencies: Data masking or obfuscation mechanisms are operational on the source ABA host OS	

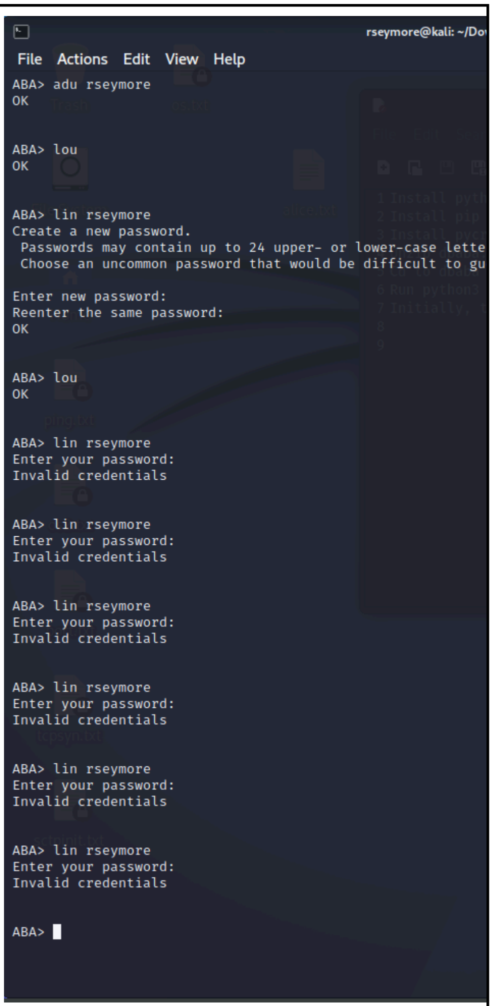
Execution

Step	Test Steps	Test Data	Notes	Expected Result	Actual Result	Status (Pass/Fail)
1	Start a shell session on the ABA host			Error message should be displayed indicating the database is encrypted when attempting to enumerate tables; Data in table rows should not be visible	All RecordIDs and associated fields/values visible using common SQL tool with simple SELECT statement	**Fail**
2	Complete the prompts to input a valid username and password	- Username: rbarber - Password: sysuserpw1				
3	Navigate to the directory where application dependencies are stored.	/home/sysuser/rbarber	Database filename can be found in unprotected file abacfg.py			
4	Test database encryption by loading the ABA database with sqlite and performing a SELECT *	- Database filename: "dbadb-db" - Default table name: "AbaTable"	Use .table to identify table name sqlite>.table sqlite> SELECT * FROM AbaTable;			

						🟥
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

Defects

Defect ID	Applicable Security Requirement	Defect Description	Steps to reproduce	Example Screenshots
1	SR-1	Minimum password requirement allows passwords that are easily crackable. Defect is due to a lack of complex password requirements integrated into the interface specification.	1. Generate a word list of all combinations of numbers, lowercase, and uppercase letters of length 1 to 8 2. Create an account with the minimum password requirements, i.e. 1 character long 3. Run each combination as an input for the password of a username	 zoemedina@Zoes-MacBook-Air dbaba % ./test.sh testing: 0 fail testing: 1 fail testing: 2 fail testing: 3 fail testing: 4 fail testing: 5 fail testing: 6 fail testing: 7 fail testing: 8 success password found: 8 zoemedina@Zoes-MacBook-Air dbaba %  zoemedina@Zoes-MacBook-Air dbaba % ./test.sh testing: 0 fail testing: 1 fail testing: 2 fail testing: 3 fail testing: 4 fail testing: 5 fail testing: 6 fail testing: 7 fail testing: 8 fail testing: 9 fail testing: a fail testing: b fail testing: c fail testing: d fail testing: e fail testing: f fail testing: g fail testing: h fail testing: i fail testing: j fail testing: k fail testing: l fail testing: m fail testing: n fail testing: o fail testing: p fail testing: q success password found: q zoemedina@Zoes-MacBook-Air dbaba %

<table>
<tr>
<td><b>2</b></td>
<td>SR-2</td>
<td>Multiple consecutive failed login attempts do not result in account lockout</td>
<td>1. Launch ABA
2. Login as admin
3. Create target account and set password
4. Attempt login to target account more than three times.
5. Note that there are no attempted login restrictions or resulting account lockouts</td>
<td>
rseymore@kali: ~/Do
ABA> adu rseymore
OK

ABA> lou
OK

ABA> lin rseymore
Create a new password.
 Passwords may contain up to 24 upper- or lower-case lette
 Choose an uncommon password that would be difficult to gu

Enter new password:
Reenter the same password:
OK

ABA> lou
OK

ABA> lin rseymore
Enter your password:
Invalid credentials

ABA> lin rseymore
Enter your password:
Invalid credentials

ABA> lin rseymore
Enter your password:
Invalid credentials

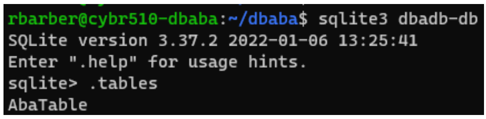
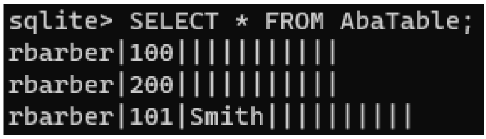
ABA> lin rseymore
Enter your password:
Invalid credentials

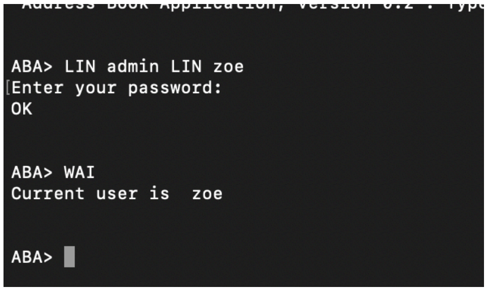
ABA> lin rseymore
Enter your password:
Invalid credentials

ABA> lin rseymore
Enter your password:
Invalid credentials

ABA> </td>
</tr>
<tr>
<td><b>3</b></td>
<td>SR-5</td>
<td>- User Records are stored in an unencrypted database accessible to system users.

- Fails to meet Security Requirement due to unauthorized disclosure of sensitive information

- Deficiency exists within the interface specification; Encryption to support SR-5 is not found during documentation review</td>
<td>1. Start a shell session on the ABA host
2. Complete the prompts to input a valid username and password
3. Navigate to directory where application dependencies are stored.
4. Load the ABA database with sqlite and list tables
5. View data in the database file using a SELECT * statement</td>
<td>
rbarber@cybr510-dbaba:~/dbaba$ sqlite3 dbadb-db
SQLite version 3.37.2 2022-01-06 13:25:41
Enter ".help" for usage hints.
sqlite> .tables
AbaTable

sqlite> SELECT * FROM AbaTable;
rbarber|100|||||||||||
rbarber|200|||||||||||
rbarber|101|Smith||||||||||</td>
</tr>
</table>

4	SR-3	- Bug identified in the LIN command permits any user with a valid login access to any other account, including the Admin user. - The interface specification is correct, but the program does not correctly implement the interface specification.	1. Launch ABA 2. Create an admin account with a password and a user account named "test" with a password 3. Enter "LIN test admin" command 4. Enter password for test account 5. Enter "WAI" command	 Address Book Application, Version 0.2 : Type ABA> LIN admin LIN zoe Enter your password: OK ABA> WAI Current user is zoe ABA>

Static Code Testing

A static code test was run on the “dbaba.py” file using the Pylint code checker. Below is the output of that test. The application was rated at **7.01/10**

```
************* Module dbaba
dbaba.py:17:138: C0303: Trailing whitespace (trailing-whitespace)
dbaba.py:17:0: C0301: Line too long (138/100) (line-too-long)
dbaba.py:66:0: C0301: Line too long (2214/100) (line-too-long)
dbaba.py:131:0: C0325: Unnecessary parens after 'return' keyword (superfluous-parens)
dbaba.py:186:17: C0303: Trailing whitespace (trailing-whitespace)
dbaba.py:217:9: C0303: Trailing whitespace (trailing-whitespace)
dbaba.py:292:17: C0303: Trailing whitespace (trailing-whitespace)
dbaba.py:337:0: C0303: Trailing whitespace (trailing-whitespace)
dbaba.py:374:0: C0301: Line too long (118/100) (line-too-long)
dbaba.py:376:0: C0301: Line too long (115/100) (line-too-long)
dbaba.py:386:0: C0325: Unnecessary parens after 'if' keyword (superfluous-parens)
dbaba.py:386:0: C0325: Unnecessary parens after 'not' keyword (superfluous-parens)
dbaba.py:408:0: C0303: Trailing whitespace (trailing-whitespace)
dbaba.py:473:0: C0301: Line too long (118/100) (line-too-long)
dbaba.py:475:0: C0301: Line too long (115/100) (line-too-long)
dbaba.py:483:0: C0325: Unnecessary parens after 'if' keyword (superfluous-parens)
dbaba.py:483:0: C0325: Unnecessary parens after 'not' keyword (superfluous-parens)
dbaba.py:493:0: C0301: Line too long (104/100) (line-too-long)
dbaba.py:499:96: C0303: Trailing whitespace (trailing-whitespace)
dbaba.py:500:21: C0303: Trailing whitespace (trailing-whitespace)
dbaba.py:503:60: C0303: Trailing whitespace (trailing-whitespace)
dbaba.py:508:66: C0303: Trailing whitespace (trailing-whitespace)
dbaba.py:517:0: C0301: Line too long (115/100) (line-too-long)
dbaba.py:554:0: C0301: Line too long (123/100) (line-too-long)
dbaba.py:554:0: C0325: Unnecessary parens after 'not' keyword (superfluous-parens)
dbaba.py:557:0: C0301: Line too long (104/100) (line-too-long)
dbaba.py:566:60: C0303: Trailing whitespace (trailing-whitespace)
dbaba.py:615:0: C0301: Line too long (123/100) (line-too-long)
dbaba.py:615:0: C0325: Unnecessary parens after 'not' keyword (superfluous-parens)
dbaba.py:618:0: C0301: Line too long (104/100) (line-too-long)
dbaba.py:627:60: C0303: Trailing whitespace (trailing-whitespace)
dbaba.py:686:87: C0303: Trailing whitespace (trailing-whitespace)
dbaba.py:697:0: C0301: Line too long (102/100) (line-too-long)
dbaba.py:707:89: C0303: Trailing whitespace (trailing-whitespace)
dbaba.py:710:0: C0301: Line too long (106/100) (line-too-long)
dbaba.py:718:0: C0301: Line too long (115/100) (line-too-long)
dbaba.py:724:0: C0301: Line too long (110/100) (line-too-long)
dbaba.py:785:0: C0303: Trailing whitespace (trailing-whitespace)
dbaba.py:804:26: C0303: Trailing whitespace (trailing-whitespace)
dbaba.py:816:0: C0325: Unnecessary parens after 'if' keyword (superfluous-parens)
dbaba.py:817:0: C0325: Unnecessary parens after 'if' keyword (superfluous-parens)
dbaba.py:819:0: C0325: Unnecessary parens after 'elif' keyword (superfluous-parens)
dbaba.py:824:0: C0325: Unnecessary parens after 'elif' keyword (superfluous-parens)
dbaba.py:829:0: C0325: Unnecessary parens after 'elif' keyword (superfluous-parens)
dbaba.py:834:0: C0325: Unnecessary parens after 'elif' keyword (superfluous-parens)
dbaba.py:837:0: C0325: Unnecessary parens after 'elif' keyword (superfluous-parens)
dbaba.py:840:0: C0325: Unnecessary parens after 'elif' keyword (superfluous-parens)
dbaba.py:843:0: C0325: Unnecessary parens after 'elif' keyword (superfluous-parens)
dbaba.py:848:0: C0325: Unnecessary parens after 'elif' keyword (superfluous-parens)
dbaba.py:854:0: C0325: Unnecessary parens after 'elif' keyword (superfluous-parens)
dbaba.py:857:0: C0325: Unnecessary parens after 'elif' keyword (superfluous-parens)
dbaba.py:860:0: C0325: Unnecessary parens after 'elif' keyword (superfluous-parens)
dbaba.py:863:0: C0325: Unnecessary parens after 'elif' keyword (superfluous-parens)
dbaba.py:865:0: C0325: Unnecessary parens after 'elif' keyword (superfluous-parens)
dbaba.py:870:0: C0325: Unnecessary parens after 'elif' keyword (superfluous-parens)
dbaba.py:875:0: C0325: Unnecessary parens after 'elif' keyword (superfluous-parens)
dbaba.py:878:0: C0325: Unnecessary parens after 'elif' keyword (superfluous-parens)
dbaba.py:881:0: C0325: Unnecessary parens after 'elif' keyword (superfluous-parens)
dbaba.py:896:0: C0303: Trailing whitespace (trailing-whitespace)
dbaba.py:897:52: C0303: Trailing whitespace (trailing-whitespace)
dbaba.py:905:0: C0303: Trailing whitespace (trailing-whitespace)
dbaba.py:906:0: C0303: Trailing whitespace (trailing-whitespace)
dbaba.py:365:34: W1401: Anomalous backslash in string: '\w'. String constant might be missing an r prefix. (anomalous-backslash-in-string)
dbaba.py:365:40: W1401: Anomalous backslash in string: '\S'. String constant might be missing an r prefix. (anomalous-backslash-in-string)
dbaba.py:464:34: W1401: Anomalous backslash in string: '\w'. String constant might be missing an r prefix. (anomalous-backslash-in-string)
dbaba.py:464:40: W1401: Anomalous backslash in string: '\S'. String constant might be missing an r prefix. (anomalous-backslash-in-string)
dbaba.py:1:0: C0114: Missing module docstring (missing-module-docstring)
dbaba.py:25:0: C0116: Missing function or method docstring (missing-function-docstring)
dbaba.py:25:0: C0103: Function name "AbaStartup" doesn't conform to snake_case naming style (invalid-name)
dbaba.py:38:0: C0116: Missing function or method docstring (missing-function-docstring)
dbaba.py:49:0: C0116: Missing function or method docstring (missing-function-docstring)
dbaba.py:49:0: C0103: Function name "GoodRecordID" doesn't conform to snake_case naming style (invalid-name)
dbaba.py:51:4: R1703: The if statement can be replaced with 'return bool(test)' (simplifiable-if-statement)
dbaba.py:51:4: R1705: Unnecessary "else" after "return", remove the "else" and de-indent the code inside it (no-else-return)
dbaba.py:64:0: C0116: Missing function or method docstring (missing-function-docstring)
dbaba.py:64:0: C0103: Function name "StrongPassword" doesn't conform to snake_case naming style (invalid-name)
dbaba.py:83:0: C0116: Missing function or method docstring (missing-function-docstring)
dbaba.py:113:0: C0116: Missing function or method docstring (missing-function-docstring)
dbaba.py:113:0: C0103: Function name "GetCommand" doesn't conform to snake_case naming style (invalid-name)
dbaba.py:120:4: W0621: Redefining name 'e' from outer scope (line 901) (redefined-outer-name)
dbaba.py:120:11: W0718: Catching too general exception Exception (broad-exception-caught)
dbaba.py:120:4: C0103: Variable name "e" doesn't conform to snake_case naming style (invalid-name)
dbaba.py:127:11: W0718: Catching too general exception Exception (broad-exception-caught)
dbaba.py:127:4: C0103: Variable name "e" doesn't conform to snake_case naming style (invalid-name)
dbaba.py:120:4: W0612: Unused variable 'e' (unused-variable)
dbaba.py:139:0: C0116: Missing function or method docstring (missing-function-docstring)
dbaba.py:165:0: C0116: Missing function or method docstring (missing-function-docstring)
dbaba.py:171:7: C0121: Comparison 'cur_usr != None' should be 'cur_usr is not None'
(singleton-comparison)
dbaba.py:190:12: W0104: Statement seems to have no effect (pointless-statement)
dbaba.py:190:12: C0121: Comparison 'first == True' should be 'first is True' if checking for the singleton
value True, or 'bool(first)' if testing for truthiness (singleton-comparison)
dbaba.py:198:11: C0121: Comparison 'first == True' should be 'first is True' if checking for the singleton
value True, or 'first' if testing for truthiness (singleton-comparison)
dbaba.py:210:0: C0116: Missing function or method docstring (missing-function-docstring)
dbaba.py:233:0: C0116: Missing function or method docstring (missing-function-docstring)
dbaba.py:250:0: C0116: Missing function or method docstring (missing-function-docstring)
dbaba.py:270:0: C0116: Missing function or method docstring (missing-function-docstring)
dbaba.py:290:0: C0116: Missing function or method docstring (missing-function-docstring)
dbaba.py:307:0: C0116: Missing function or method docstring (missing-function-docstring)
dbaba.py:311:4: C0103: Variable name "targetUsr" doesn't conform to snake_case naming style
(invalid-name)
dbaba.py:321:12: C0103: Variable name "targetUsr" doesn't conform to snake_case naming style
(invalid-name)
dbaba.py:330:0: C0116: Missing function or method docstring (missing-function-docstring)
dbaba.py:343:0: C0116: Missing function or method docstring (missing-function-docstring)
dbaba.py:365:16: R1723: Unnecessary "else" after "break", remove the "else" and de-indent the code
inside it (no-else-break)
dbaba.py:343:0: R0912: Too many branches (16/12) (too-many-branches)
dbaba.py:414:0: C0116: Missing function or method docstring (missing-function-docstring)
dbaba.py:439:0: C0116: Missing function or method docstring (missing-function-docstring)
dbaba.py:464:16: R1723: Unnecessary "else" after "break", remove the "else" and de-indent the code
inside it (no-else-break)
dbaba.py:499:20: W0104: Statement seems to have no effect (pointless-statement)
dbaba.py:503:20: C0200: Consider using enumerate instead of iterating with range and len
(consider-using-enumerate)
dbaba.py:482:4: R1702: Too many nested blocks (6/5) (too-many-nested-blocks)
dbaba.py:515:24: W0106: Expression "completion_code == db.Insert_record(list(tmprecorddict.values()))"
is assigned to nothing (expression-not-assigned)
dbaba.py:517:24: W0104: Statement seems to have no effect (pointless-statement)
dbaba.py:439:0: R0912: Too many branches (23/12) (too-many-branches)
dbaba.py:441:4: W0612: Unused variable 'emptyparamdict' (unused-variable)
dbaba.py:529:0: C0116: Missing function or method docstring (missing-function-docstring)
dbaba.py:575:16: C0206: Consider iterating with .items() (consider-using-dict-items)
dbaba.py:529:0: R0912: Too many branches (13/12) (too-many-branches)
dbaba.py:594:0: C0116: Missing function or method docstring (missing-function-docstring)
dbaba.py:636:16: C0206: Consider iterating with .items() (consider-using-dict-items)
dbaba.py:659:0: C0116: Missing function or method docstring (missing-function-docstring)
dbaba.py:680:8: W0621: Redefining name 'e' from outer scope (line 901) (redefined-outer-name)
dbaba.py:680:15: W0718: Catching too general exception Exception (broad-exception-caught)
dbaba.py:677:22: W1514: Using open without explicitly specifying an encoding (unspecified-encoding)
dbaba.py:680:8: C0103: Variable name "e" doesn't conform to snake_case naming style (invalid-name)
dbaba.py:691:12: R1724: Unnecessary "else" after "continue", remove the "else" and de-indent the code
inside it (no-else-continue)
dbaba.py:708:20: R1723: Unnecessary "else" after "break", remove the "else" and de-indent the code
inside it (no-else-break)
dbaba.py:708:23: R1714: Consider merging these comparisons with 'in' by using 'rdcompletion_code not
in ('OK', 'No record found')'. Use a set instead if elements are hashable. (consider-using-in)
dbaba.py:710:24: W0104: Statement seems to have no effect (pointless-statement)
dbaba.py:718:32: W0104: Statement seems to have no effect (pointless-statement)
dbaba.py:684:4: R1702: Too many nested blocks (7/5) (too-many-nested-blocks)
dbaba.py:724:28: W0104: Statement seems to have no effect (pointless-statement)
dbaba.py:659:0: R0912: Too many branches (19/12) (too-many-branches)
dbaba.py:684:4: R1702: Too many nested blocks (6/5) (too-many-nested-blocks)
dbaba.py:677:22: R1732: Consider using 'with' for resource-allocating operations (consider-using-with)
dbaba.py:664:4: W0612: Unused variable 'tmprecordlist' (unused-variable)
dbaba.py:739:0: C0116: Missing function or method docstring (missing-function-docstring)
dbaba.py:760:8: W0621: Redefining name 'e' from outer scope (line 901) (redefined-outer-name)
dbaba.py:760:15: W0718: Catching too general exception Exception (broad-exception-caught)
dbaba.py:759:22: W1514: Using open without explicitly specifying an encoding (unspecified-encoding)
dbaba.py:760:8: C0103: Variable name "e" doesn't conform to snake_case naming style (invalid-name)
dbaba.py:777:27: W0718: Catching too general exception Exception (broad-exception-caught)
dbaba.py:777:20: C0103: Variable name "e" doesn't conform to snake_case naming style (invalid-name)
dbaba.py:759:22: R1732: Consider using 'with' for resource-allocating operations (consider-using-with)
dbaba.py:769:20: W0612: Unused variable 'discard' (unused-variable)
dbaba.py:794:0: C0116: Missing function or method docstring (missing-function-docstring)
dbaba.py:800:8: R1722: Consider using 'sys.exit' instead (consider-using-sys-exit)
dbaba.py:823:16: R1722: Consider using 'sys.exit' instead (consider-using-sys-exit)
dbaba.py:794:0: R0912: Too many branches (30/12) (too-many-branches)
dbaba.py:794:0: R0915: Too many statements (88/50) (too-many-statements)
dbaba.py:904:4: R1722: Consider using 'sys.exit' instead (consider-using-sys-exit)

------------------------------------------------------------------
Your code has been rated at 7.01/10 (previous run: 7.01/10, +0.00)
```