



Protocol Audit Report

Version 1.0

Cyfrin.io

January 6, 2024

Protocol Audit Report

Owen Lee

January 6, 2024

Prepared by: Owen Lee Lead Security Researcher: - Owen Lee

Table of Contents

- Table of Contents
- Protocol Summary
- Disclaimer
- Risk Classification
- Audit Details
 - Scope
 - Roles
- Executive Summary
 - Issues found
- Findings
 - High
 - * [H-1] Storing the password on-chain makes it visible to anyone, and no longer private.
 - * [H-2] `PasswordStore::setPassword` has no access control, meaning a non-owner can change the password.
 - Informational
 - * [I-1] The `PasswordStore::getPassword` natspec indicates a parameter that doesn't exist, causing the natspec to be incorrect.

Protocol Summary

PasswordStore is a protocol dedicated to storage and retrieval of a user’s password. The protocol is designed to be used by a single user, and is not designed to be used by multiple users. Only the owner should be able to set and access this password.

Disclaimer

The YOUR_NAME_HERE team makes all effort to find as many vulnerabilities in the code in the given time period, but holds no responsibilities for the findings provided in this document. A security audit by the team is not an endorsement of the underlying business or product. The audit was time-boxed and the review of the code was solely on the security aspects of the Solidity implementation of the contracts.

Risk Classification

		Impact		
		High	Medium	Low
Likelihood	High	H	H/M	M
	Medium	H/M	M	M/L
	Low	M	M/L	L

We use the CodeHawks severity matrix to determine severity. See the documentation for more details.

Audit Details

The findings described in this document correspond to the following commit hash:

```
1 2e8f81e263b3a9d18fab4fb5c46805ffc10a9990
```

Scope

```
1 ./src/  
2 #-- PasswordStore.sol
```

Roles

- Owner: The user who can set the password and read the password.
- Outsiders: No one else should be able to set or read the password. # Executive Summary

**Add some notes about how the audit went, types of things you found, etc. We spent X hours with Z auditors using Y tools. etc*

Issues found

Severity	Number of issues found
High	2
Medium	0
Low	0
Info	1
Gas Optimizations	0
Total	3

Findings

High

[H-1] Storing the password on-chain makes it visible to anyone, and no longer private.

Description: All data stored on-chain is visible to anyone, and can be read directly on the blockchain. The `PasswordStore::s_password` variable is intended to be a private variable and only accessed through the `PasswordStore::getPassword` function, which is intended to be only called by the owner of the contract.

We show one such method of reading any data off chain below.

Proof of Concept: (Proof of Code)

1. Create a locally running chain

2. Deploy the contract to the chain

3. Run the storage tool

```
1 function setPassword(string memory newPassword) external {
2   @> // @audit - There are no access controls here
3     s_password = newPassword;
4     emit SetNetPassword();
5 }
```

Impact: Anyone can set/change the password of the contract.

Proof of Concept:

Add the following to the `PasswordStore.t.sol` test suite.

Code

```
1 function test_anyone_can_set_password(address randomAddress) public {
2   vm.prank(randomAddress);
3   string memory expectedPassword = "myNewPassword";
4   passwordStore.setPassword(expectedPassword);
5   vm.prank(owner);
6   string memory actualPassword = passwordStore.getPassword();
7   assertEquals(actualPassword, expectedPassword);
8 }
```

Recommended Mitigation: Add an access control modifier to the `setPassword` function.

```
1 if (msg.sender != s_owner) {
2   revert PasswordStore__NotOwner();
3 }
```

Informational

[I-1] The PasswordStore::getPassword natspec indicates a parameter that doesn't exist, causing the natspec to be incorrect.

Description:

```
1 /*
2  * @notice This allows only the owner to retrieve the password.
3  * @param newPassword The new password to set.
4  */
5 function getPassword() external view returns (string memory) {
```

The natspec for the function `PasswordStore::getPassword` indicates it should have a parameter with the signature `getPassword(string)`. However, the actual function signature is `getPassword()`.

Impact: The natspec is incorrect.

Recommended Mitigation: Remove the incorrect natspec.

```
1 - * @param newPassword The new password to set.
```