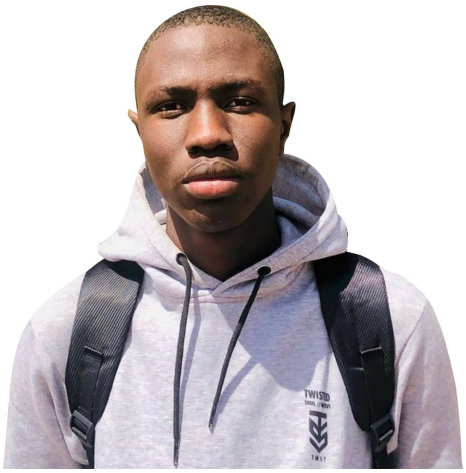
# Protocol Audit Report

Mustapha Abdulaziz Sani (musty_code.py)

03 9, 2024



# Protocol Audit Report

Version 1.0

*Mustapha Abdulaziz Sani*

September 3, 2024

# Protocol Audit Report

Mustapha Abdulaziz Sani (musty_code.py)

03 9, 2024

Prepared by: Mustapha Abdulaziz Sani Lead Auditors: - xxxxxxx

## Table of Contents



## Protocol Summary

PasswordStore is a protocol dedicated to storage and retrieval's of user's passwords.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. Protocol does X, Y, Z

## 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 |
| | High | H | H/M | M |
| Likelihood | 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 the following commit hash**

2e8f81e263b3a9d18fab4fb5c46805ffc10a9990

### Scope

- In Scope:

```
./src/
--PasswordStore.sol
```

### Roles

- Onwer: 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 summary likes tools, hours spent and etc

### Issues found

| Severity | Number of issues found |
|---|---|

High | 2 |
Medium | 0 |
low | 0 |
informational | 1 |
Total | 3 |

# Findings

# High

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

**Description:** All data stored on-chain is visible to anyone, and can be read directly from the blockchain. The PasswordStore::s_password variable is inteed to be a private variable and only accessrd 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

**Impact:** Anyone can read the private password, severly breaking the functionality of the protocol

**Proof of Concept:**

**Recommended Mitigation:** Due to this, the overall architechture of the contract should be rethrought. one could encrypt the password. this would require the user to remember another password off-chain to decrypt the password. However, you'd also likely want to remove the view function as you would'nt want user to accidentally send a transaction with the password that encrypts your password

### [H-2] `PasswordStore::setPasswordhas` no access controls, meaning non-onwer can change password

**Description:** The PasswordStore::setPassword function is set to be an external function however, the natspec of the function and overall purpose of smart contracts is that This function allows only the owner to set a new password

```
function setPassword(string memory newPassword) external {
    //@audit:- there are no access controls
        s_password = newPassword;
        emit SetNetPassword();
    }
```

**Impact:** Anyone can set/change password of control, severely breaking the control intended functionality

**Proof of Concept:**Add the following to the passwordStore.t.soltest file.

```
    function test_anyone_can_set_password(address randomAddress) public {
        vm.assume(randomAddress != owner);
        vm.prank(randomAddress);
        string memory expectedPassword = "myNewPassword";
        passwordStore.setPassword(expectedPassword);
        vm.prank(owner);
        string memory  actualPassword = passwordStore.getPassword();
        assertEq(actualPassword, expectedPassword);
    }
```

**Recommended Mitigation** Add an access control conditional to the setPassword function.

```
if(msg.sender != s_owner) {
    revert passwordStore__NotOwner();
}
```

```
# Informational
### [I-1] The `passwordStore:getPassword` natspec indicates a parameter that doe

**Description:**
```javascript
 /*
 *@notice this allows only the owner to retrieve the password
 *@param newPassword the new password to set
 */
function getPassword() external view returns (string memory){

}
```

the passwordStore::getPassword function signature is getPassword() which the natspec say it should be getPassword(string).

**Impact:** The natspec is incorrect

**Proof of Concept:**

**Recommended Mitigation:** remove the incorrect natspec line

```
-    @param newPassword the new password to set
```
```