# Protocol Audit Report

Version 1.0

*Emmanuel*

May 19, 2024

# Protocol Audit Report

Emmanuel

May 17, 2023

Prepared by: Emmanuel Lead Auditors: Emmanuel - xxxxxxx

## Table of Contents

## Protocol Summary

PasswordStore is a protocol dedicated to storage and retrieval of a 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.

## Disclaimer

The Emmanuel 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 in this document correspond with the following commit hash:** Commit Hash:

```
7d55682ddc4301a7b13ae9413095feffd9924566
```

### Scope

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

### Roles

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

## Executive Summary

*we found some really interesting things. Spent 4hrs auditing the code.

### Issues found

| Severity | Number of issues found |
|---|---|
| High | 2 |
| Medium | 0 |
| Low | 0 |
| Info | 0 |
| Total | 2 |

## Findings

### High

#### [H-1] Variables stored in storage on-chin are visible to anyone, no matter the solidity visibility keyword. Meaning the password is not actually 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 intended to be a private variable and only be accessed through the `PasswordStore::getPassword` function, which is intended to be called by the owner of the contract.

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

**Proof of Concept:** (Proof of code)

The below test case shows how anyone can read the password directly from the blockchain.

1. Create a locally running chain

```
make anvil
```

2. Deploy the contract to the chain

```
make deploy
```

3. Run the strorage tool

I use `1` because that is the storage slot of `s_password` in the contract.

```
cast storage --rpc-url http://127.0.0.1.8545 <ADDRESS_HERE> 1
```

I got an output that looks like this: `0x6456954300000000000000000000000000000000000000000000000014`

I parse that hex to a string with:

```
cast parse-bytes32-string 0
    x6456954300000000000000000000000000000000000000000000000014
```

And get an output of:

```
myPassword
```

**Recommended Mitigation:**

### [H-2] `PassowrdStore::setPassword` has no access comtrol, meaning a non-owner could change the password.

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

**Impact:** Anyone can set/change the password of the contract, severly breaking the intended functionality of the contract

**Proof of Concept:** Add the following to the `PasswordStore.t.sol` test 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 conditioner to the `setPassword` function. “‘javascript

pandoc report.md -o report.pdf –from markdown –template=eisvogel –listings