Protocol Audit Report

Version 1.0

Cyfrin.io

September 5, 2025

Protocol Audit Report

Dominick

September 5 2025

Prepared by: Dominick Lead Auditors: - xxxxxxx

Table of Contents

Protocol Summary

PasswordStore is a protocol dedicated to stoarage and retreival ofa user's password. The protocol is designed to be used by a single user, and not be used by multiple users. Only the owner should be able to set and access this password.

Disclaimer

The InlineSix 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

Commit hash

```
2e8f81e263b3a9d18fab4fb5c46805ffc10a9990
```

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

Issues found

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

Findings

High

[H-1] Variables stored on-chain are visible to anyone, no matter the visibility.

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 accessed through the `PasswordStore::s_getPassword` function.

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

Impact: Anyonbe can read the private password, severely breaking the functionality of the protocol.

Proof of Concept:

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 storage tool

```
cast storage <ADDRESS_HERE>
```

You'll get an ouput that looks like this:

```
s_password: 0
    x6d7950617373776f726400000000000000000000000000000000000000000014
```

You can parse that hex into a string with:

```
cast parse-bytes32-string 0
    x6d7950617373776f726400000000000000000000000000000000000000000014
```

And get an output of:

```
myPassword
```

Recommended Mitigation: Due to this, the overall archeticture of the contract should be rethought. One could encrypt the password off-chain, and then store the encrypted password on-chain. This woould require the user to remember another password off-chain to decrypt the password.

[H-2] `PasswordStore::setPassword` has no access controls, meaning a non-owner can change the password.

Informational

[I-1] The `PasswordStore::getPassword` natspec indicates a parameter that doesnt exist