Protocol Audit Report

Mineter

May 8, 2025

Protocol Audit Report

Version 1.0

Client: 3-passwordstore

Auditor: Mineter

May 9, 2025

Protocol Audit Report

Prepared by: Mineter Lead Auditor: - Mineter

Table of Contents

Protocol Summary

A smart contract application for storing a password.The protocol is designed to be used by a single user. Users should be able to store a password and then retrieve it later. Others should not be able to access the password.

Disclaimer

The Mineter’s 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 corespond the following commit hash

`0x5FbDB2315678afecb367f032d93F642f64180aa3`

Scope

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

Roles

- Owner: The user who set the password, only person that should have acces to a pasword
- Outsides: No one else should have acces to set or read the password

Executive Summary

We spent 1 hour on this protocol with only 1 auditor - used tools like a foundry - to spot a vulnerabilities. Our time was allocated appropriately to the code base.

Issues found

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

Findings

High

[H-1] Variable "s_password" declared as private, is not really private - anyone can read

Description: All data stored on-chain are visible to anyone, and can be read directly from the blockchain. The `PasswordStore::s_password` 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 private password, severly breaking the functionality of the protocol.

Proof of Concept:

1. Start a locally running chain

   ```
   make anvil
   ```

2. Deploy contrat to that chain

   ```
   make deploy
   ```

3. Getting a variable ***s_password***

```
cast storage <transaction hash> 1
```

4. Transfer to a readible text

```
cast parse-bytes32-string  <output from step 3>
```

5. Secret password is ours

Recommended Mitigation: Due to this, the overall architecture of the contract should be rethought. One could encrypt the password off-chain, and then store the encrypted password on-chain. 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 wouldn't want the user to accidentally send a transaction with the password that decrypts your password.

[H-2] `PasswordStore::setPassword` is callable by anyone

Description: This function is used to set a password. That makes this function critical for security of this contract. Only owner of the contract should have acces to this.
`But there is no protection, anyone can change the password`

Impact: Anyone can change password of the contract.

Proof of Concept:

```
    function test_set_pass_as_outside(address random_addr) public {
        vm.assume(random_addr != owner);
        vm.prank(random_addr);
        targetContract.setPassword("IamNotTheOwner");

        vm.prank(owner);
        string memory supernewpass = targetContract.getPassword();
        console.log(supernewpass);
    }
```

Recommended Mitigation: Due to this, the overall architecture of the contract should be rethought. One could encrypt the password off-chain, and then store the encrypted password on-chain. 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 wouldn't want the user to accidentally send a transaction with the password that decrypts your password.