Protocol Audit Report

Version 1.0

swarecito

March 9, 2025

Protocol Audit Report

swarecito

March 9, 2025

Prepared by: Swarecito Lead Auditors:

- swarecito

Table of Contents

Protocol Summary

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
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 the following commit hash:

```
2e8f81e263b3a9d18fab4fb5c46805ffc10a9990
```

Scope

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

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.

Roles

- Owner: Is the only one who should be able to set and access the password.

For this contract, only the owner should be able to interact with the contract.

Executive Summary

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 everyone and no longer private.

Description: All data stored on-chain is visible to everyone, and can be read by anyone. The `PasswordStore::s_password` variable is intended to be a private variable and only accessed through.

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

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

Proof of Concept:

1. Run the anvil node

```
make anvil
```

2. Deploy the contract

```
make deploy
```

3. Use cast `call` to read the contract storage slot

```
cast storage 0x5FbDB2315678afecb367f032d93F642f64180aa3  1 --rpc-url
    127.0.0.1:8545 | cast --parse-bytes32-string
Output: myPassword
```

Recommended Mitigation: The recommended mitigation is to store the password off-chain, hash it, and store the hash on-chain.

[H-2] Missing access control on `PasswordStore::setPassword` allows anyone to change the password.

Description: The `PasswordStore::setPassword` function does not have any access control, allowing anyone to change the password. This is a critical vulnerability as it allows anyone to change the password, breaking the functionality of the protocol.

Impact: Anyone can change the password, severely breaking the functionality of the protocol.

Proof of Concept:

1. Run the anvil node

```
make anvil
```

2. Deploy the contract

```
make deploy
```

3. Use cast `send` to make a transaction to change the password

```
cast send 0x5FbDB2315678afecb367f032d93F642f64180aa3 "setPassword(string)" "newPassword"  --rpc-url 127.0.0.1:8545 --account password-store
```

Output:

```
blockHash               0xed294339213c91c457c3006dfc5eb834206279f72ccfa07bf96203050b2f92cd
blockNumber             3
contractAddress
cumulativeGasUsed       28290
effectiveGasPrice       768110363
from                    0xf39Fd6e51aad88F6F4ce6aB8827279cffFb92266
gasUsed                 28290
logs                    [{"address":"0x5fbdb2315678afecb367f032d93f642f64180aa3","topics":["0x2d90f9f3484ad964397201b45d21cb99b91c92f18f4417b4597ee71ebb055762"],"data":"0x","blockHash":"0xed294339213c91c457c3006dfc5eb834206279f72ccfa07bf96203050b2f92cd","blockNumber":"0x3","blockTimestamp":"0x67cdcbd3","transactionHash":"0xba825230a184b8ec712b1de55fd59d1422b3011a4089ca7ff36f8e737f69c210","transactionIndex":"0x0","logIndex":"0x0","removed":false}]
logsBloom               0x0000000000000000000000000000000000000000000000000000000000000000000000000000000
root
status                  1 (success)
transactionHash         0xba825230a184b8ec712b1de55fd59d1422b3011a4089ca7ff36f8e737f69c210
transactionIndex        0
type                    2
blobGasPrice            1
blobGasUsed
authorizationList
to                      0x5FbDB2315678afecb367f032d93F642f64180aa3
```

Recommended Mitigation: The recommended mitigation is to add access control to the `PasswordStore::setPassword` function. This can be done by adding a modifier that checks if the sender is the owner of the contract. Or adding a check in the function to ensure that only the owner can change the password.

```
function setPassword(string memory newPassword) external  {
    if (msg.sender != s_owner) {
        revert PasswordStore__NotOwner();
    }
    s_password = newPassword;
    emit SetNewPassword();
}
```

Informational

[I-1] The `PasswordStore::getPassword` function does not have any paramater whereas comments says it does which is confusing.

Description: The `PasswordStore::getPassword` function signature is `function getPassword()external view returns (string memory)` but the natspec say it is `function getPassword(string memory newPassword)external view returns (string memory)`. This is confusing and can lead to confusion.

Impact: The natspec is incorrect.

Recommended Mitigation: Update the natspec to match the function signature.

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

—### [I-1] The `PasswordStore::getPassword` function does not have any paramater whereas comments says it does which is confusing.

Description: The `PasswordStore::getPassword` function signature is `function getPassword()external view returns (string memory)` but the natspec say it is `function getPassword(string memory newPassword)external view returns (string memory)`. This is confusing and can lead to confusion.

Impact: The natspec is incorrect.

Recommended Mitigation: Update the natspec to match the function signature.

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