Protocol Audit Report

Version 1.0

Chris

September 6, 2024

Protocol Audit Report

Christopher Fernandez

September 6, 2024

Prepared by: Christopher Fernandez Lead Auditors Research: - xxxxxxx

Table of Contents

    * [I-1] The `PasswordStore::getPassword` natspec indicates a parameter that doesn't exist, causing the natspec to be incorrect.
  - Likehood & Impact:

Protocol Summary

En teoria el codigo lo que hace es guardar la contraseña de una persona especifica y solo el owner puede setear una nueva.

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

- Commit Hash: 2e8f81e263b3a9d18fab4fb5c46805ffc10a9990

- Solc Version: 0.8.18
- Chain(s) to deploy contract to: Ethereum ## Scope

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

Roles

Executive Summary

Issues found

Findings

High

[H-1]Todo lo que almacenamos en el storage es completamente publico en la blockchain - Hight

Description

Cualquier valor en el storage es completamente publico en la blockchain, por ende aunque tenga la intencion de poner el “private” en la variable seguira siendo publico en la blockchain, el private solo significa que esta restringido a este contrato y no puede ser llamada dicha variable desde otro.

`PasswordStore:: s_password` se cree que es privada y solo se puede ser accesada a trave de `PasswordStore:: getPassword`

Impact: Cualquier puede leer la contraseña **Proof of Concept;** (proof of code)

El Test de abajo demuestra que cualquier persona puede setear una contraseña

we need a local chain running.

```
anvil
```

> Note: Most PoC’s won’t require a local blockchain

Next we need to deploy our protocol, fortunately, PasswordStore has a `make` command set up for us. Note that their deploy script is setting the password `myPassword` in the process. Open a new terminal and run the following.

```
make deploy
```

Foundry allows us to check the storage of a deployed contract with a very simple `cast` command. For this we'll need to recall to which storage slot the `s_password` variable is assigned.

With this consideration we can run the command `cast storage <address> <storageSlot>` like this (*your address may be different*).

here the storage slot is 1 because the owner is slot 0

```
cast storage <ADDRESS_HERE> 1 --rpc-url http://127.0.0.1:8545
```

We should receive an output similar to this:

```
`0x6d7950617373776f726400000000000000000000000000000000000000000014`
```

This is the bytes form of the data at `storage slot 1`. By using another convenient Foundry command we can now decode this data.

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

Our output then becomes:

```
myPassword
```

And we've done it. In a few quick commands we've shown that the data our client is expecting to keep hidden on chain is accessible to anyone. Let's add these steps as proof to our report. Things are getting long, so I've collapsed the report examples going forward!

Recommended Mitigation:

Debido a esto, la arquitectura general del contrato debe reconsiderarse. Se podría cifrar la contraseña fuera de la cadena y luego almacenar la contraseña cifrada en la cadena. Esto requeriría que el usuario recordara otra contraseña fuera de la cadena para descifrar la contraseña almacenada. Sin embargo, es probable que también desee eliminar la función de visualización, ya que no querría que el usuario enviara accidentalmente una transacción con esta clave de descifrado.

Likehood & Impact:

- Impact: High
- Likelihood: High
- Severity: High

[H-2] `PasswordStore::SetPassword` has no access control, meaning now-owner could change the password

Description

En `PasswordStore::SetPassword` dicha funcion es external pero ademas la funcion que se pide es que `solo el owner pueda cambiar la contraseña`

```
1          function setPassword(string memory newPassword) external {
2          // @audit aqui no hay control de acceso
3          s_password = newPassword;
4          emit SetNetPassword();
5      }
```

Impact:

Anyone can set/change the password, severily breaking the contract intented functionality

Proof of Concept;

Add the following to the PasswordStore.t.sol test file:

```
1  function test_anyone_can_set_password(address randomAddress) public {
2          vm.assume(randomAddress != owner);
3          vm.startPrank(randomAddress);
4          string memory expectedPassword = "myNewPassword";
5          passwordStore.setPassword(expectedPassword);
6
7          vm.startPrank(owner);
8          string memory actualPassword = passwordStore.getPassword();
9          assertEq(actualPassword, expectedPassword);
10     }
```

Recommended Mitigation:

Add an access control conditional to `PasswordStore::setPassword`.

```
1  if(msg.sender != s_owner){
2      revert PasswordStore__NotOwner();
3  }
```

Likehood & Impact:

- Impact: HIGHT
- Likelihood: HIGHT
- Severity: HIGHT

Informational

[I-1] The `PasswordStore::getPassword` natspec indicates a parameter that doesn't exist, causing the natspec to be incorrect.

Description: ”’ / @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()` while the natspec says it should be `getPassword(string)`.

Impact The natspec is incorrect

Recommended Mitigation: Remove the incorrect natspec line

```
1      /*
2       * @notice This allows only the owner to retrieve the password.
3  -    * @param newPassword The new password to set.
4       */
```

Likehood & Impact:

- Impact: None
- Likelihood: None
- Severity: Informational