Protocol Audit Report

Version 1.0

Jaunepr

2024 年 5 月 25 日

Protocol Audit Report

Jaunepr

May 24, 2024

Prepared by: Jaunepr Lead Auditors: - Jaunepr

Table of Contents

Protocol Summary

Protocol description blablabla…

Disclaimer

(Blablabla) 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 the document correspond the following commit hash:

```
Commit Hash: 2e8f81e263b3a9d18fab4fb5c46805ffc10a9990
```

Scope

```
./src/
—— PasswordStore.sol
```

Roles

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

Executive Summary

审查过程摘要 我们花费了多少小时，使用了 *Foundry* 测试框架，*etc..*

Issues found

严重程度	问题个数
高	2
中	0
低	0
提示	1
共计	3

Findings

High

[S-1] 密码存储在（storage）中，在链上是对所有人公开的

描述（Description): 所有存储在 storage 中的数据，在链上是对所有人公开的，可以直接从链上获取。变量`PasswordStore::s_password`应该是一个私有变量只能由合约拥有者通过`PasswordStore::getPassword`函数读取。

下面我会展示一种从链下读取任何数据的示例。

影响 (Impact): 任何人都可以访问读取密码，严重破坏了协议的功能。

Proof of Concept:

```
下面是一个测试案例，可以证明任何人可以直接从链上读取数据。
```

1. 创建并运行一个本地测试区块链

```
anvil
```

2. 部署合约

```
make deploy
```

3. 运行 storage 工具使用1因为`password`存储在 storage 插槽 1

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

然后你会得到如下输出:`0x6d7950617373776f7264000000000000000000000000000000000000000000000014`

4. 将上面的十六进制结果转换成字符串

```
cast parse-bytes32-string 0
    x6d7950617373776f7264000000000000000000000000000000000000000000000014
```

然后你会得到如下输出：`myPassword`

缓解措施 (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 stored password. However, you’re also likely want to remove the view function as you wouldn’t want the user to accidentally send a transaction with this decryption key. 基于上述问题，应该重新考虑合约的整体架构。建议方法一：可以将密码在链下加密之后存储到链上。这将需要合约拥有者链外拥有解密方法，或另一个密码来解密。建议方法二：可以将 view 函数移除，防止用户意外使用密钥查看密码。

[S-2] `PasswordStore::setPassword` has no access controls, means non-owner could change the password.

描述 (Description): `PasswordStore::setPassword`被设置为了`external`函数，而该函数功能与其智能合约的目的是`该函数只允许拥有者设置密码`

```
    function setPassword(string memory newPassword) external {
@>      // @audit: there are no access controls.
        s_password = newPassword;
        emit SetNetPassword();
    }
```

影响 (Impact): 任何人可以修改和改变已设置的密码，甚至可能破坏智能合约的意向功能。

Proof of Concept: 添加以下代码到`PasswordStore.t.sol`测试文件

code

```
    function test_anyone_can_set_password(address randomAddress) public {
        vm.assume(randomAddress != owner);
        vm.prank(randomAddress);
        string memory newPassword = "hahaIsNew";
        passwordStore.setPassword(newPassword);

        vm.prank(owner);
        string memory actualPassword = passwordStore.getPassword();
        assertEq(actualPassword, newPassword);
    }
```

缓解措施 (Recommended Mitigation): 添加如下访问控制代码到`PasswordStore.sol::setPassword`中

```
if(msg.sender != owner){
    revert PasswordStore__NotOwner();
}
```

Informational

[I-1] TITLE (Root Cause + Impact) `PasswordStore::getPassword` natspec indicated a parameter named newPassword that doesn’t exist. casue natspec to be incorrect.

描述 (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) {
```

按上面注释要求，`getPassword()`应该是`getPassword(string)`

影响 (Impact): The natspec is incorrect.

缓解措施 (Recommended Mitigation): 移除错误的注释

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