PasswordStore Initial Audit Report

SnavOhBurmaa

February 18, 2026

Contents

1 About Me

Hi, I'm new.

2 Disclaimer

The Ohburmaa 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.

3 Risk Classification

	High Impact	Medium Impact	Low Impact
High Likelihood	H	H/M	M
Medium Likelihood	H/M	M	M/L
Low Likelihood	M	M/L	L

4 Audit Details

Commit Hash Reviewed:

`2e8f81e263b3a9d18fab4fb5c46805ffc10a9990`

4.1 Scope

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

5 Protocol Summary

PasswordStore is designed for storage and retrieval of a user's password. The contract is intended for single-user usage. Only the owner should be able to set and access the password.

5.1 Roles

Owner: The only address allowed to set and retrieve the password.

6 Executive Summary

Severity	Number of Issues
High	2
Medium	0
Low	1
Info	1
Gas	0

7 Findings

[H-1] Password Stored On-Chain Is Publicly Visible

Description:

All blockchain data is publicly accessible. Even if a variable is marked `private`, it can still be read directly from contract storage using off-chain tools.

Impact:

The password is not private.

Proof of Concept:

```
cast storage <ADDRESS> 1
cast parse-bytes32-string <HEX_OUTPUT>
```

Recommendation:

Do not store plaintext passwords on-chain. Instead:

- Store a hash of the password
- Or encrypt off-chain and store encrypted value
- Remove unnecessary view functions that expose secrets

[H-2] setPassword Callable by Anyone

Description:

The function does not restrict access:

```
function setPassword(string memory newPassword) external {
    s_password = newPassword;
}
```

Impact:

Anyone can change the password.

Recommendation:

Add access control:

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

[L-01] Initialization Timeframe Vulnerability

Summary:

The password is not initialized in the constructor. During deployment, it defaults to an empty string.

Impact:

Temporary unintended state.

Recommendation:

Set the password in the constructor.

[I-1] Incorrect NatSpec Documentation

Description:

The NatSpec comment indicates a parameter that does not exist in `getPassword()`.

Impact:

Documentation mismatch.

Recommendation:

Remove the incorrect `@param` line.

8 Conclusion

The contract contains critical architectural issues regarding password storage and access control. Design improvements are required before production use.