Prepared by: KineticsOfWeb3
Lead Smart Contract Security Researcher:

Table of Contents

Protocol Summary

PasswordStore Smart Contract allows users to securely store and retrieve their passwords while ensuring that no unauthorized individuals can access them.

- **Password Storage**: Users can store their passwords within the smart contract. This password is associated with the user's account and is intended to remain private.
- **Password Retrieval**: Users can later retrieve the stored password by calling a retrieval function. Access to the password is restricted to the user who initially stored it.
- **Access Control**: Only the user who set the password can retrieve it. No other users, even if they interact with the contract, should be able to view or modify the stored password.

Disclaimer

KineticsOfWeb3 makes all efforts to find as many vulnerabilities in the code in the given time period, but holds no responsibility for the findings provided in this document. A security audit by me is not an endorsement of the underlying business or product. The audit was time-boxed, and the review of the code was solely focused 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

The CodeHawks severity matrix was used to determine severity. See the documentation for more details.

Audit Details

The findings described in this document correspond to the following commit hash:

```
2e8f81e263b3a9d18fab4fb5c46805ffc10a9990
```

Scope

```
./src/
└── PasswordStore.sol
```

Roles

Roles Owner:

The individual who deploys the contract or the account that stores the password. Has the exclusive ability to set and retrieve the stored password. Responsible for maintaining their private key, which is required to interact with the contract securely. Non-Owner/Unauthorized Users:

Any account or individual that is not the owner. They are unable to view or modify the password stored by the owner. Unauthorized users attempting to retrieve the password will be blocked by the access control mechanisms.

Executive Summary

This report outlines the results of the audit on the **PasswordStore** smart contract. The contract allows users to store and retrieve passwords securely, but several vulnerabilities were found during the audit:

- **High-risk issue**: Missing access control, allowing any user to set passwords.
- **Minor issues**: Incorrect NatSpec documentation and lack of password confidentiality, as private variables are still accessible on-chain.

A total of **10 hours** were spent auditing the codebase, performing tests, and writing a Proof of Concept (PoC) to demonstrate exploitability. The key vulnerabilities were classified and mitigation strategies were proposed.

With proper fixes, the contract's security and reliability can be significantly improved before moving to production.

Issues found

Severity	Issue Description	Details	Status
High	Missing Access Control	Any user can set the password, leading to unauthorized manipulation of the stored password.	Unresolved
High	Password Visibility (Blockchain Privacy Issue)	Although marked private, passwords stored in the contract can be viewed on-chain.	Unresolved
Medium	Incorrect NatSpec Documentation	The NatSpec indicates a parameter that doesn't exist, leading to confusion for developers.	Unresolved
Informational	No low severity findings	None identified during the audit.	N/A

Total Findings

Severity	Count
High	2
Medium	1
Informational	1
Low	0