soken

SMART CONTRACT SECURITY AUDIT

Snowflake (SNOW)

Scan and check this report
was posted at Soken Github

November, 2022

Website: soken.io

Table of Contents

Disclaimer

This is a comprehensive report based on our automated and manual examination of cybersecurity vulnerabilities and framework flaws of the project's smart contract.

Reading the full analysis report is essential to build your understanding of project's security level. It is crucial to take note, though we have done our best to perform this analysis and report, that you should not rely on the our research and cannot claim what it states or how we created it.

Before making any judgments, you have to conduct your own independent research.

We will discuss this in more depth in the following disclaimer - please read it fully.

DISCLAIMER: You agree to the terms of this disclaimer by reading this report or any portion thereof. Please stop reading this report and remove and delete any copies of this report that you download and/or print if you do not agree to these conditions. Scan and verify report's presence in the GitHub repository by a qr-code at the title page. This report is for non-reliability information only and does not represent investment advice. No one shall be entitled to depend on the report or its contents, and Soken and its affiliates shall not be held responsible to you or anyone else, nor shall Soken provide any guarantee or representation to any person with regard to the accuracy or integrity of the report.

Without any terms, warranties or other conditions other than as set forth in that exclusion and Soken excludes hereby all representations, warrants, conditions and other terms (including, without limitation, guarantees implied by the law of satisfactory quality, fitness for purposes and the use of reasonable care and skills).

The report is provided as "as is" and does not contain any terms and conditions. Except as legally banned, Soken disclaims all responsibility and responsibilities and no claim against Soken is made to any amount or type of loss or damages (without limitation, direct, indirect, special, punitive, consequential or pure economic loses or losses) that may be caused by you or any other person, or any damages or damages, including without limitations (whether innocent or negligent).

Security analysis is based only on the smart contracts. No applications or operations were reviewed for security. No product code has been reviewed.

Procedure

Our analysis contains following steps:

1. Project Analysis;

2. Manual analysis of smart contracts:
   - Deploying smart contracts on any of the network(Ropsten/Rinkeby) using Remix IDE
   - Hashes of all transaction will be recorded
   - Behaviour of functions and gas consumption is noted, as well.

3. Unit Testing:
   - Smart contract functions will be unit tested on multiple parameters and under multiple conditions to ensure that all paths of functions are functioning as intended.
   - In this phase intended behaviour of smart contract is verified.
   - In this phase, we would also ensure that smart contract functions are not consuming unnecessary gas.
   - Gas limits of functions will be verified in this stage.

4. Automated Testing:
   - Mythril
   - Oyente
   - Manticore
   - Solgraph

Terminology

We categorize the finding into 4 categories based on their vulnerability:

- Low-severity issue — less important, must be analyzed
- Medium-severity issue — important, needs to be analyzed and fixed
- High-severity issue —important, might cause vulnerabilities, must be analyzed and fixed
- Critical-severity issue —serious bug causes, must be analyzed and fixed.

Limitations

The security audit of Smart Contract cannot cover all vulnerabilities. Even if no vulnerabilities are detected in the audit, there is no guarantee that future smart contracts are safe. Smart contracts are in most cases safeguarded against specific sorts of attacks. In order to find as many flaws as possible, we carried out a comprehensive smart contract audit. Audit is a document that is not legally binding and guarantees nothing.

Basic Security Recommendation

Unlike hardware and paper wallets, hot wallets are connected to the internet and store private keys online, which exposes them to greater risk. If a company or an individual holds significant amounts of cryptocurrency in a hot wallet, they should consider using MultiSig addresses. Wallet security is enhanced when private keys are stored in different locations and are not controlled by a single entity.

More info: **https://blog.soken.io/how-to-gnosis-multisig-1c6c0860586f**

Token Contract Details for 31.10.2022

Contract Name: **Token**

Deployed address: **0xE0f463832295ADf63eB6CA053413a3f9cd8bf685**

Total Supply: **200,000,000**

Token Tracker: **SNOW**

Decimals: **18**

Token holders: **6**

Transactions count: **6**

Top 100 holders dominance: **100%**

Audit Details

Project Name: **Snowflake**

Language: **Solidity**

Compiler Version: **v0.8.9**

Blockchain: **Polygon**

Social Profiles

Project Website: **In development**

Project Twitter: **In development**

Project Telegram: **https://t.me/snowflake_exchange**

Vulnerabilities checking

Issue Description	Checking Status
Compiler Errors	Completed
Delays in Data Delivery	Completed
Re-entrancy	Completed
Transaction-Ordering Dependence	Completed
Timestamp Dependence	Completed
Shadowing State Variables	Completed
DoS with Failed Call	Completed
DoS with Block Gas Limit	Completed
Outdated Complier Version	Completed
Assert Violation	Completed
Use of Deprecated Solidity Functions	Completed
Integer Overflow and Underflow	Completed
Function Default Visibility	Completed
Malicious Event Log	Completed
Math Accuracy	Completed
Design Logic	Completed
Fallback Function Security	Completed
Cross-function Race Conditions	Completed
Safe Zeppelin Module	Completed

Security Issues

1) Incorrect Access Control: Medium-severity L575-578, L594-597, L611-L615, L629-632, L648-651, L969-L971, L982-L1023

Access control plays an important role in segregation of privileges in smart contracts and other applications. If this is misconfigured or not properly validated on sensitive functions, it may lead to loss of funds, tokens and in some cases compromise of the smart contract. The contract Token is importing an access control library @openzeppelin/contracts/access/Ownable.sol but the function delegateBySig is missing the modifier onlyOwner.

2) Presence of Overpowered Role (Mint function): Informational L908-L912

The overpowered owner (i.e., the person who has too much power) is a project design where the contract is tightly coupled to their owner (or owners); only they can manually invoke critical functions. Due to the fact that this function is only accessible from a single address, the system is heavily dependent on the address of the owner. In this case, there are scenarios that may lead to undesirable consequences for investors, e.g., if the private key of this address is compromised, then an attacker can take control of the contract.

Conclusion for project owner

Medium and informational-severity issues exist within smart contracts.

NOTE: Please check the disclaimer above and note, that audit makes no statements or warranties on business model, investment attractiveness or code sustainability. Contract security report for community

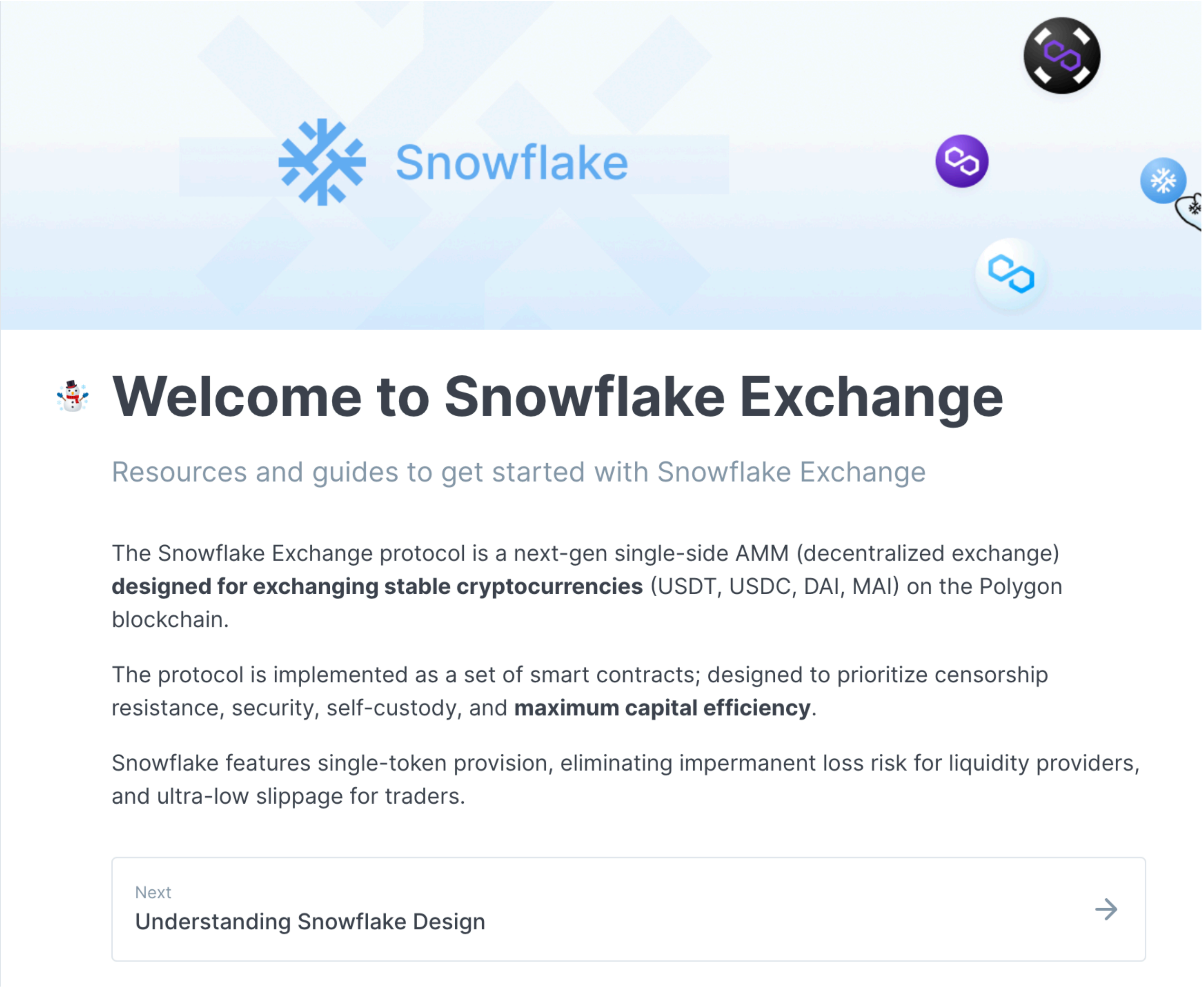
SECURITY REPORT
FOR COMMUNITY

SNOW

soken

Whitepaper of the project

The whitepaper of Snowflake project has been verified on behalf of Soken team.



⛄ Welcome to Snowflake Exchange

Resources and guides to get started with Snowflake Exchange

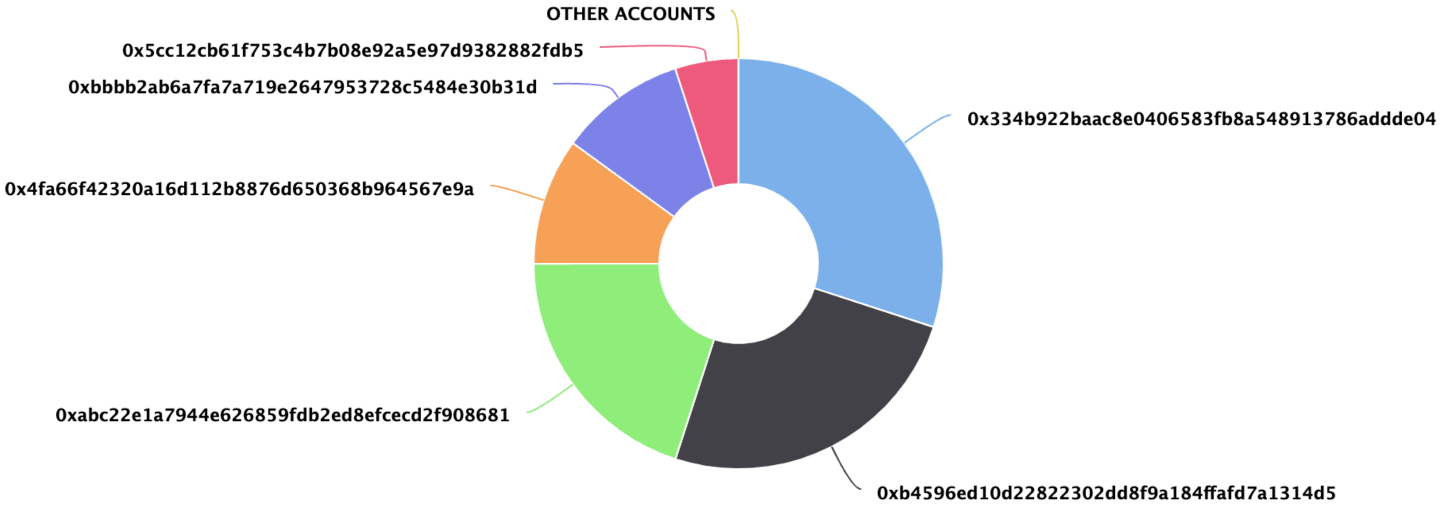
The Snowflake Exchange protocol is a next-gen single-side AMM (decentralized exchange) **designed for exchanging stable cryptocurrencies** (USDT, USDC, DAI, MAI) on the Polygon blockchain.

The protocol is implemented as a set of smart contracts; designed to prioritize censorship resistance, security, self-custody, and **maximum capital efficiency**.

Snowflake features single-token provision, eliminating impermanent loss risk for liquidity providers, and ultra-low slippage for traders.

Whitepaper link: **https://snowflake-exchange.gitbook.io/snowflake-exchange/**

SNOW Token Distribution



SNOW Top 10 Holders

Rank	Address	Quantity (Token)	Percentage
1	0x334b922baac8e0406583fb8a548913786addde04	60,000,000	30.0000%
2	0xb4596ed10d22822302dd8f9a184ffafd7a1314d5	50,000,000	25.0000%
3	0xabc22e1a7944e626859fdb2ed8efcecd2f908681	40,000,000	20.0000%
4	0x4fa66f42320a16d112b8876d650368b964567e9a	20,000,000	10.0000%
5	0xbbbb2ab6a7fa7a719e2647953728c5484e30b31d	20,000,000	10.0000%
6	0x5cc12cb61f753c4b7b08e92a5e97d9382882fdb5	10,000,000	5.0000%

Soken Contact Info

Website: www.soken.io

Mob: (+1)416-875-4174

32 Britain Street, Toronto, Ontario, Canada

Telegram: @team_soken

GitHub: sokenteam

Twitter: @soken_team