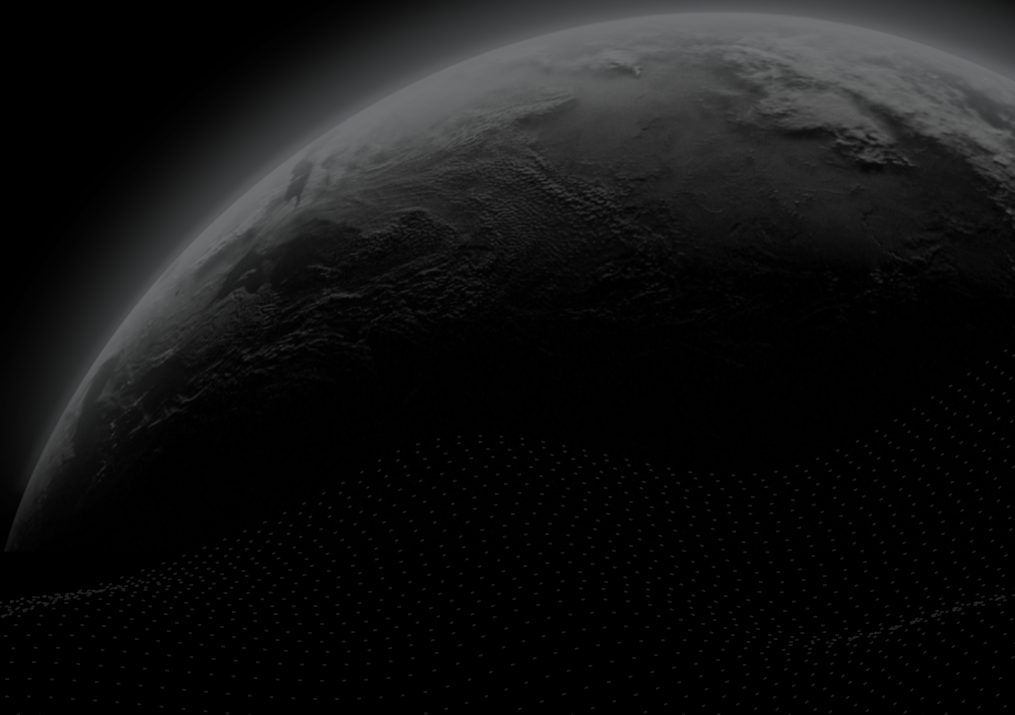




Security Assessment

STFX

CertiK Verified on Dec 12th, 2022





CertiK Verified on Dec 12th, 2022

STFX

The security assessment was prepared by CertiK, the leader in Web3.0 security.

Executive Summary

TYPES

Platform

ECOSYSTEM

Ethereum

METHODS

Manual Review, Static Analysis

LANGUAGE

Solidity

TIMELINE

Delivered on 12/12/2022

KEY COMPONENTS

N/A

CODEBASE

<https://github.com/STFX-IO/stfx-single-contract/tree/8f77134dea22477c18f8ff6f96bf300f6c5f83df>
<https://github.com/STFX-IO/tokenomics->
[...View All](#)

COMMITTS

8f77134dea22477c18f8ff6f96bf300f6c5f83df
 06269de309d4768fedd3b1f29166b691e611938e
[...View All](#)

Vulnerability Summary



16

Total Findings

3

Resolved

0

Mitigated

0

Partially Resolved

13

Acknowledged

0

Declined

0

Unresolved

0 Critical

Critical risks are those that impact the safe functioning of a platform and must be addressed before launch. Users should not invest in any project with outstanding critical risks.

1 Major

1 Acknowledged



Major risks can include centralization issues and logical errors. Under specific circumstances, these major risks can lead to loss of funds and/or control of the project.

1 Medium

1 Acknowledged



Medium risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform.

5 Minor

5 Acknowledged



Minor risks can be any of the above, but on a smaller scale. They generally do not compromise the overall integrity of the project, but they may be less efficient than other solutions.

9 Informational

3 Resolved, 6 Acknowledged



Informational errors are often recommendations to improve the style of the code or certain operations to fall within industry best practices. They usually do not affect the overall functioning of the code.

TABLE OF CONTENTS | STFX

I **Summary**

[Executive Summary](#)

[Vulnerability Summary](#)

[Codebase](#)

[Audit Scope](#)

[Approach & Methods](#)

I **Review Notes**

[Financial Models](#)

I **Findings**

[STF-01 : Centralization Related Risks](#)

[STF-02 : Unchecked Value of ERC-20 `transfer\(\)`/`transferFrom\(\)` Call](#)

[STX-01 : Third Party Dependencies](#)

[STX-02 : Modifier `onlyManager` not used](#)

[SVS-01 : Missing process of flag `closed`](#)

[SVS-02 : Incompatibility With Deflationary Tokens](#)

[SVS-03 : admin operations](#)

[RSF-01 : Missing Emit Events](#)

[RST-01 : Logical issue of function `getPrice\(\)` in `perp/Reader.sol`](#)

[SSF-01 : Redundant comment](#)

[SVS-04 : Logical issue in function `depositInfoFund\(\)`](#)

[SVS-05 : Checks on hardcoded chainId](#)

[SVS-06 : Logical issue of function `cancelOrder\(\)`](#)

[SVS-07 : Logical issue of function `createNewStf\(\)`](#)

[SVS-08 : Logical issue of creating stf](#)

[SVS-09 : Logical issue about leverage](#)

I **Appendix**

I **Disclaimer**

CODEBASE | STFX

Repository

<https://github.com/STFX-IO/stfx-single-contract/tree/8f77134dea22477c18f8ff6f96bf300f6c5f83df>

<https://github.com/STFX-IO/tokenomics-contracts/tree/06269de309d4768fedd3b1f29166b691e611938e>








Commit

8f77134dea22477c18f8ff6f96bf300f6c5f83df

06269de309d4768fedd3b1f29166b691e611938e

AUDIT SCOPE | STFX

7 files audited ● 7 files with Acknowledged findings

ID	File	SHA256 Checksum
● STF	 src/STFXToken.sol	ce76f3126d3751c3f6178564121e7559143b60890a0defef3b7019e39e76c5b0
● PST	 src/Presale.sol	15e2702638465ef9c194852793aa4efcc868b4f468429d2a6acd17b3dda28167
● SVS	 src/StfxVault.sol	9051ca222acb48daf1bd6cb01b501ae02edcd69162d392b01d154c2a28f61496
● RST	 src/perp/Reader.sol	581e4096872b16b65f8f688cc9713f1d483b44e6fd89f64f47c8c87e55ace571
● SST	 src/perp/Stfx.sol	b1072fb9b88b7d1d422e0bad47d56d4c33c7e66767140dd76f1f2d8070ad5f36
● RSF	 src/gmx/Reader.sol	662ddd250283e461f3a21133eb38d4d918e7568c43f4171e25456c25fb16ff76
● SSF	 src/gmx/Stfx.sol	ed416f9e88db62114398802e1172f76c891619615349a1bea402c5f19fee0eb2

APPROACH & METHODS | STFX

This report has been prepared for STFX to discover issues and vulnerabilities in the source code of the STFX project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Manual Review and Static Analysis techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

The security assessment resulted in findings that ranged from critical to informational. We recommend addressing these findings to ensure a high level of security standards and industry practices. We suggest recommendations that could better serve the project from the security perspective:

- Testing the smart contracts against both common and uncommon attack vectors;
- Enhance general coding practices for better structures of source codes;
- Add enough unit tests to cover the possible use cases;
- Provide more comments per each function for readability, especially contracts that are verified in public;
- Provide more transparency on privileged activities once the protocol is live.

REVIEW NOTES | STFX

The STFX protocol is a Defi investment system that allows fund managers to raise funds and invest in representing the users.

I Financial Models

The STFX protocol allows users to become the stf manager and raise money from the community to open positions in ClearingHouse protocol or GMX protocol.

The manager makes an investment decision and users can choose one manager they trust to get involved in the investment. So, the users should accept any risks and losses during the process.

Financial models of blockchain protocols need to be resilient to attacks. They need to pass simulations and verifications to guarantee the security of the overall protocol.

The financial model of this protocol is not in the scope of this audit.

FINDINGS | STFX



16

Total Findings

0

Critical

1

Major

1

Medium

5

Minor

9

Informational

This report has been prepared to discover issues and vulnerabilities for STFX. Through this audit, we have uncovered 16 issues ranging from different severity levels. Utilizing the techniques of Manual Review & Static Analysis to complement rigorous manual code reviews, we discovered the following findings:

ID	Title	Category	Severity	Status
<u>STF-01</u>	Centralization Related Risks	Centralization / Privilege	Major	● Acknowledged
<u>STF-02</u>	Unchecked Value Of ERC-20 <code>transfer()</code> / <code>transferFrom()</code> Call	Volatile Code	Minor	● Acknowledged
<u>STX-01</u>	Third Party Dependencies	Volatile Code	Minor	● Acknowledged
<u>STX-02</u>	Modifier <code>onlyManager</code> Not Used	Logical Issue	Minor	● Acknowledged
<u>SVS-01</u>	Missing Process Of Flag <code>closed</code>	Logical Issue	Medium	● Acknowledged
<u>SVS-02</u>	Incompatibility With Deflationary Tokens	Logical Issue	Minor	● Acknowledged
<u>SVS-03</u>	Admin Operations	Control Flow	Minor	● Acknowledged
<u>RSF-01</u>	Missing Emit Events	Coding Style	Informational	● Acknowledged
<u>RST-01</u>	Logical Issue Of Function <code>GetPrice()</code> In <code>perp/Reader.sol</code>	Logical Issue	Informational	● Acknowledged
<u>SSF-01</u>	Redundant Comment	Coding Style	Informational	● Acknowledged

ID	Title	Category	Severity	Status
<u>SVS-04</u>	Logical Issue In Function <code>depositInfoFund()</code>	Logical Issue	Informational	● Resolved
<u>SVS-05</u>	Checks On Hardcoded ChainId	Logical Issue	Informational	● Acknowledged
<u>SVS-06</u>	Logical Issue Of Function <code>cancelOrder()</code>	Logical Issue	Informational	● Acknowledged
<u>SVS-07</u>	Logical Issue Of Function <code>createNewStf()</code>	Logical Issue	Informational	● Resolved
<u>SVS-08</u>	Logical Issue Of Creating Stf	Logical Issue	Informational	● Resolved
<u>SVS-09</u>	Logical Issue About Leverage	Logical Issue	Informational	● Acknowledged

STF-01 | CENTRALIZATION RELATED RISKS

Category	Severity	Location	Status
Centralization / Privilege	● Major	src/StfxVault.sol (8f77134dea22477c18f8ff6f96bf300f6c5f83df); src/gmx/Reader.sol (8f77134dea22477c18f8ff6f96bf300f6c5f83df); src/Presale.sol (06269de309d4768fedd3b1f29166b691e611938e); src/STFXToken.sol (06269de309d4768fedd3b1f29166b691e611938e)	● Acknowledged

Description

In the contract **gmx/Reader.sol**, the role **owner** has authority over the following functions:

- function setDex(), to set `dex` address.
- function setOwner(), to change the owner of the contract.

Any compromise to the **owner** account may allow a hacker to take advantage of this authority and change the config for the protocol.

In the contract **StfxVault.sol**, the role **admin** has authority over the following functions:

- function closePosition(), to close the stfx position.
- function cancelOrder(), to cancel the pending order.
- function distributeProfits(), to distribute profits to users.
- function closeLiquidatedVault(), to close the stfx liquidated by the dex.
- function cancelVault(), to cancel the empty or expired stfx.
- function cancelStfAfterOpening(), to cancel the pending stfx.
- function cancelStfAfterPositionDeadline(), to cancel the stf after a month and close the position as a market order.
- function pause() and unpause(), to set the pause status.

AND

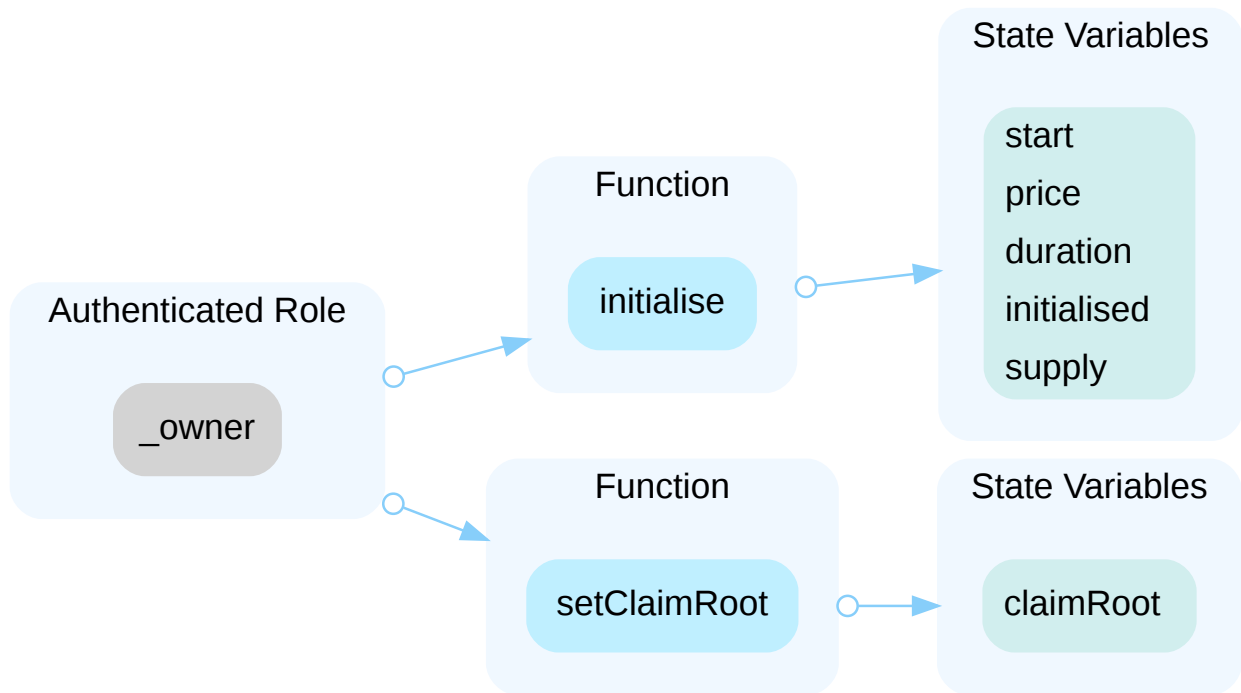
The role `owner` has authority over the following functions:

- function setCapacityPerStf(), to set the max capacity of collateral which can be raised per stf.
- function setMinInvestmentAmount(), to set the min investment of collateral an investor can invest per stf.
- function setMaxInvestmentAmount() and setMaxLeverage(), to set max leverage and invest amount.
- function setMinLeverage(), to set min leverage.

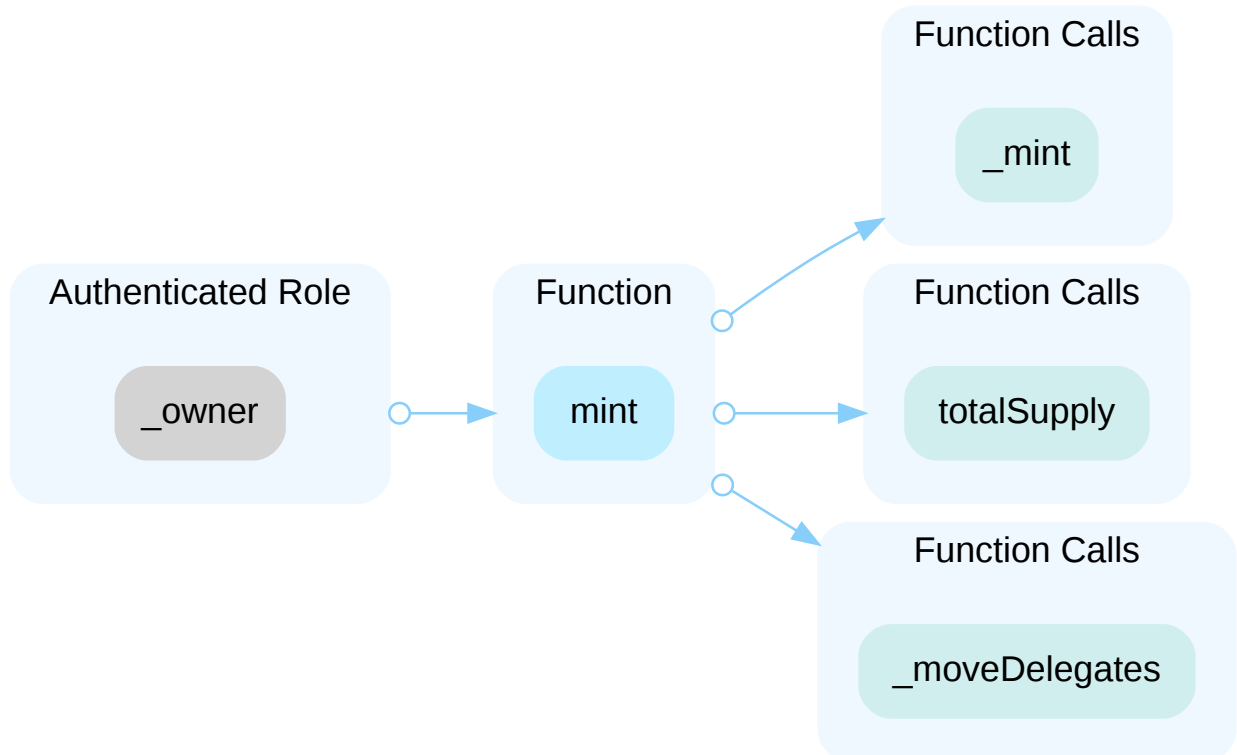
- function `setMaxFundraisingPeriod()`, to set the max fundraising period.
- function `setMaxDeadlineForPosition()`, to set the max deadline a position can be open for an stf.
- function `setManagerFee()`, to set the manager fee percent.
- function `setProtocolFee()`, to set the protocol fee percent.
- function `setOwner()`, to set owner.
- function `setStfxImplementation()`, to set the new stfx implementation.
- function `setReader()`, to set the reader.
- function `setFundDeadline()`, to set the fundDeadline.
- function `setUsdc()`, to set usdc address.
- function `setWeth()`, to set weth address.
- function `setAdmin()`, to set admin.
- function `setReferrerCode()`, to set referralCode.
- function `setIsManagingFund()`, to set the isManagingFund.
- function `withdraw()`, `withdrawToken()`, `withdrawFromStf()`, to withdraw fund from the contracts.

Any compromise to the **owner** and **admin** accounts may allow a hacker to take advantage of this authority and bring unpredictable damages to the project.

In the contract `Presale` the role `_owner` has authority over the functions shown in the diagram below. Any compromise to the `_owner` account may allow the hacker to take advantage of this authority and bring unpredictable damages to the project.



In the contract `STFXToken` the role `_owner` has authority over the functions shown in the diagram below. Any compromise to the `_owner` account may allow the hacker to take advantage of this authority and bring unpredictable damages to the project.



Recommendation

The risk describes the current project design and potentially makes iterations to improve in the security operation and level of decentralization, which in most cases cannot be resolved entirely at the present stage. We recommend carefully managing the privileged account's private key to avoid any potential risks of being hacked. In general, we strongly recommend centralized privileges or roles in the protocol be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., multi-signature wallets.

Indicatively, here are some feasible suggestions that would also mitigate the potential risk at a different level in terms of short-term, long-term and permanent:

Short Term:

Timelock and Multi sign ($\frac{2}{3}$, $\frac{3}{5}$) combination *mitigate* by delaying the sensitive operation and avoiding a single point of key management failure.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
AND
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key compromised;
AND
- A medium/blog link for sharing the timelock contract and multi-signers addresses information with the public audience.

Long Term:

Timelock and DAO, the combination, *mitigate* by applying decentralization and transparency.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
AND
- Introduction of a DAO/governance/voting module to increase transparency and user involvement;
AND
- A medium/blog link for sharing the timelock contract, multi-signers addresses, and DAO information with the public audience.

Permanent:

Renouncing the ownership or removing the function can be considered *fully resolved*.

- Renounce the ownership and never claim back the privileged roles;
OR
- Remove the risky functionality.

I Alleviation

The team acknowledged the issue and stated the following:

"Our `owner` address will be a multi-sig. And our `admin` address will be the one which is used by our backend bot. Since it has to be active and keep calling contract functions to cancel and close the stfs, unfortunately, we can't use a multi-sig for that. But we are actively looking into other options to avoid any centralization risks."

STF-02 | UNCHECKED VALUE OF ERC-20 `transfer()` / `transferFrom()` CALL

Category	Severity	Location	Status
Volatile Code	Minor	src/StfxVault.sol (8f77134dea22477c18f8ff6f96bf300f6c5f83df): 372, 413, 444, 607, 944; src/gmx/Stfx.sol (8f77134dea22477c18f8ff6f96bf300f6c5f83df): 271, 306~307, 316, 329; src/perp/Stfx.sol (8f77134dea22477c18f8ff6f96bf300f6c5f83df): 134~135, 142; src/Presale.sol (06269de309d4768fedd3b1f29166b691e611938e): 128	Acknowledged

Description

The linked `transfer()` / `transferFrom()` invocations do not check the return value of the function call, which should yield `true` in the case of proper ERC-20 implementation.

Recommendation

Since some ERC-20 tokens return no values and others return a `bool` value, they should be handled with care. We recommend using OpenZeppelin's [SafeERC20.sol](#) implementation to interact with the `transfer()` and `transferFrom()` functions of external ERC-20 tokens. The OpenZeppelin implementation checks for the existence of a return value and reverts if `false` is returned, making it compatible with all ERC-20 token implementations.

Alleviation

The team acknowledged the issue and stated the following:

"We will fix the issue in the future, which will not be included in this audit engagement."

STX-01 | THIRD PARTY DEPENDENCIES

Category	Severity	Location	Status
Volatile Code	Minor	src/gmx/Reader.sol (8f77134dea22477c18f8ff6f96bf300f6c5f83df); src/gmx/Stfx.sol (8f77134dea22477c18f8ff6f96bf300f6c5f83df); src/perp/Reader.sol (8f77134dea22477c18f8ff6f96bf300f6c5f83df); src/perp/Stfx.sol (8f77134dea22477c18f8ff6f96bf300f6c5f83df)	Acknowledged

Description

The contract is serving as the underlying entity to interact with 3rd party **ClearingHouse** protocols. The scope of the audit treats 3rd party entities as black boxes and assumes their functional correctness. However, in the real world, 3rd parties can be compromised and this may lead to lost or stolen assets. Additionally, upgrades of 3rd parties can possibly create severe impacts, such as increasing fees of 3rd parties, migrating to new LP pools, etc.

In file `gmx/Reader.sol` :

- `IGmxVault(dex.vault).shortableTokens()` (line 49)
- `IGmxVaultPriceFeed(vaultPriceFeed).getPrice()` (line 59)

In file `gmx/Stfx.sol` :

- `IGmxPositionRouter().minExecutionFee()` (line 123, 194, 249)
- `IGmxRouter().approvePlugin()` (line 138, 210)
- `IGmxOrderBook().createIncreaseOrder()` (line 139)
- `IGmxPositionRouter().createIncreasePosition()` (line 154)
- `IGmxOrderBook().createDecreaseOrder()` (line 211)
- `IGmxPositionRouter().createDecreasePosition()` (line 217)
- `IGmxOrderBook().cancelIncreaseOrder()` (line 246)
- `IGmxOrderBook().cancelDecreaseOrder()` (line 259)
- `IGmxReader().getAmountOut()` (line 338, 350)
- `IGmxRouter().swap()` (line 357)

In file `perp/Reader.sol` :

- `IMarketRegistry().hasPool()` (line 23)
- `IBaseToken().getIndexPrice()` (line 35)

In file `perp/Stfx.sol` :

- **IClearingHouse().** `openPosition()` (line 77, 91)
- **IclearingHouse().** `closePosition()` (line 111)

In file `StfxVault.sol` :

- **IGmxVault().** `getPosition()` (line 230)

Recommendation

We understand that the business logic of **Stfx** requires interaction with **ClearingHouse**, etc. We encourage the team to constantly monitor the statuses of 3rd parties to mitigate the side effects when unexpected activities are observed.

Alleviation

The team acknowledged the issue and stated the following:

"Since we are dependant on third party protocols, we'll make sure we constantly monitor their statuses."

STX-02 | MODIFIER `onlyManager` NOT USED

Category	Severity	Location	Status
Logical Issue	● Minor	src/StfxVault.sol (8f77134dea22477c18f8ff6f96bf300f6c5f83df): 334; src/gmx/Stfx.sol (8f77134dea22477c18f8ff6f96bf300f6c5f83df): 70	● Acknowledged

Description

The modifier `onlyManager` is not used in the contract.

Recommendation

We recommend the team remove the role if it is redundant.

Alleviation

The team acknowledged the issue and stated the following:

"We will fix the issue in the future, which will not be included in this audit engagement."

SVS-01 | MISSING PROCESS OF FLAG `closed`

Category	Severity	Location	Status
Logical Issue	● Medium	src/StfxVault.sol (8f77134dea22477c18f8ff6f96bf300f6c5f83df): 482	● Acknowledged

I Description

The bool variable `closed` is defaulted to false. And when `block.chainid == 10`, it is not set to true. So the status of the `stf` is not changed after the function `closePosition()` is called.

I Recommendation

We recommend the team check the logic and fix the issue.

I Alleviation

The team acknowledged the issue and stated that they will fix in the new version.

SVS-02 | INCOMPATIBILITY WITH DEFLATIONARY TOKENS

Category	Severity	Location	Status
Logical Issue	● Minor	src/StfxVault.sol (8f77134dea22477c18f8ff6f96bf300f6c5f83df): 36 8~372	● Acknowledged

Description

The contract `vault` receives collateral tokens from the user and allows the manager to invest. The collateral token should not be deflationary tokens because the amount recorded is not the really received amount but the parameter amount (before tax). When transferring standard ERC20 deflationary tokens, the input amount may not be equal to the received amount due to the charged (and burned) transaction fee. As a result, this may not meet the assumption behind these low-level asset-transferring routines and will bring unexpected balance inconsistencies.

Recommendation

We recommend the team avoid using deflationary collateral tokens. As it is the users themselves to become managers, better to record actually received amount in the function rather than the input value `amount`.

Alleviation

The team acknowledged the issue and stated the following:

"For now, we are using major stable coins as collateral and not planning to use any deflationary ERC20 tokens as collateral. But we will definitely consider this and change it in the future if required."

SVS-03 | ADMIN OPERATIONS

Category	Severity	Location	Status
Control Flow	● Minor	src/StfxVault.sol (8f77134dea22477c18f8ff6f96bf300f6c5f83df): 68 5	● Acknowledged

Description

We understand that if anyone can become a manager, the project team should have control over managers. When the admin receives complaints and decides to take over the stf, the admin himself/herself may not represent the wishes of every investor that trusts the manager. Investors join the manager's stf, which means they trust the manager. But the admin has not been able to obtain the approval of investors, as long as the admin interferes with the stfx, there is bound to be a part of the users opposed.

Recommendation

We would like the team to elaborate more on the issue and how the project team can be a decent administrator.

Alleviation

The team acknowledged the issue and stated the following:

"Our backend bot uses the admin address and is required to cancel and close stfs. For now, it cancels an stf if there was no money raised during the fundraising period or if the manager fails to open a position 72 hours after funds were raised. And it also closes a position if it has been open for more than 30 days.

Also, since gmx executes their trades in 2 txs, our bot distributes the collateral received after closing a position on gmx depending on if its a profit or a loss."

RSF-01 | MISSING EMIT EVENTS

Category	Severity	Location	Status
Coding Style	● Informational	src/gmx/Reader.sol (8f77134dea22477c18f8ff6f96bf300f6c5f83df): 94, 102	● Acknowledged

Description

Functions that update state variables should emit relevant events as notifications.

Recommendation

We recommend adding events for state-changing actions, and emitting them in their relevant functions.

```
102     event SetOwner(address owner);
103     function setDex(Gmx calldata _dex) external {
104         require(msg.sender == owner, "Not owner");
105         dex = _dex;
106         emit SetOwner(dex);
107     }
```

Alleviation

The team acknowledged the issue and stated the following:

"We will fix the issue in the future, which will not be included in this audit engagement."

RST-01 | LOGICAL ISSUE OF FUNCTION GETPRICE() IN perp/Reader.sol

Category	Severity	Location	Status
Logical Issue	● Informational	src/perp/Reader.sol (8f77134dea22477c18f8ff6f96bf300f6c5f83df): 26	● Acknowledged

| Description

According to the logic of the protocol, the function `getPrice()` should return the instant price of the `baseToken`. While the price is not from Price Oracle but the function of the `baseToken` itself. Since the implementation of `baseToken` is not found in scope. We hope the team to elaborate more on the price consult mechanism of `baseToken`.

| Recommendation

We recommend the team elaborate on the issue.

| Alleviation

The team stated the following:

"For our current design, we are depending on the third party protocols (Gmx, Perp), where we trade, for the price of the `baseToken`, but later we'll implement our own price mechanism."

SSF-01 | REDUNDANT COMMENT

Category	Severity	Location	Status
Coding Style	● Informational	src/gmx/Stfx.sol (8f77134dea22477c18f8ff6f96bf300f6c5f83df): 122	● Acknowledged

Description

The comment at line 122 is redundant because it describes the condition that could happen in closing positions but it appears in the function `openPosition()`. And it is not aligned with the surrounding codes.

Recommendation

We recommend the team remove the comment.

Alleviation

The team acknowledged the issue and stated the following:

"We will fix the issue in the future, which will not be included in this audit engagement."

SVS-04 | LOGICAL ISSUE IN FUNCTION `depositInfoFund()`

Category	Severity	Location	Status
Logical Issue	● Informational	src/StfxVault.sol (8f77134dea22477c18f8ff6f96bf300f6c5f83df): 3 61~364	● Resolved

Description

The comment of the function `depositIntoFund()` says that amount has to be between `minInvestmentAmount` and `maxInvestmentAmount`. But the implementation seems not to align with the design.

As per the implementation, `minInvestmentAmount` represents for the lower limit of one investment. However, `maxInvestmentAmount` represents the upper limit of the total investments.

Recommendation

We would like to confirm with the client if the current implementation aligns with the original project design.

Alleviation

The team explained that `maxInvestmentAmount` represents the max investment per investor per stf. And `capacityPerStf` represents the max capacity per stf.

SVS-05 | CHECKS ON HARDCODED CHAINID

Category	Severity	Location	Status
Logical Issue	● Informational	src/StfxVault.sol (8f77134dea22477c18f8ff6f96bf300f6c5f83df): 415~419, 477~483	● Acknowledged

Description

According to the logic, the protocol on different chains will interact with different `stfx` implementations. While it is not recommended to use the check in hard code. It's recommended to directly convert the address to `IStfxGmx` or `IStfxPerp` in constructor of the contract.

Recommendation

We recommend the team check the logic and avoid using hardcoded.

Alleviation

The team acknowledged the issue and stated they will look into a way to avoid using hardcoded chainID.

SVS-06 | LOGICAL ISSUE OF FUNCTION `cancelOrder()`

Category	Severity	Location	Status
Logical Issue	● Informational	src/StfxVault.sol (8f77134dea22477c18f8ff6f96bf300f6c5f83df): 501~511	● Acknowledged

Description

According to the comment of the function `cancelOrder()`, the parameter `_isOpen` is used to decide which type of order to close. When `_isOpen` is false, it's assumed that the function should revert if the status is OPENED.

Recommendation

We would like to confirm with the client if the check aligns with the original project design.

Alleviation

The team acknowledged the issue and stated the following:

"Yes. It should revert if the status is OPENED. But we are planning to introduce partial closing where the status is still OPENED even if there's a close limit order created. Will definitely consider this when updating to include partial closing."

SVS-07 | LOGICAL ISSUE OF FUNCTION `createNewStf()`

Category	Severity	Location	Status
Logical Issue	● Informational	src/StfxVault.sol (8f77134dea22477c18f8ff6f96bf300f6c5f83df): 315	● Resolved

Description

The managers have to raise money before opening the position. But if they raised no money or a little, it might be a waste of gas to open a position or create the `stf` contract. So it might be better to create the `stf` contract when the manager is really about to invest.

Recommendation

We recommend the team elaborate more on the issue.

Alleviation

The team acknowledged the issue and stated the following:

"Our design is intended in a way where anyone can create an STF with a particular fundraising period and does not need the manager to invest in their own stf. If the stf raises money, then the manager can trade, and if not, then our bot cancels the stf after the fundraising period ends."

SVS-08 | LOGICAL ISSUE OF CREATING STF

Category	Severity	Location	Status
Logical Issue	● Informational	src/StfxVault.sol (8f77134dea22477c18f8ff6f96bf300f6c5f83df): 315	● Resolved

Description

We understand that the protocol needs to publicize the manager's investment strategy at the beginning of the creation, which is beneficial to gain the trust of users. While the users can copy the strategy from the events and open positions themselves to avoid fees, which might be unfair to those experienced managers.

Recommendation

We would like the team to elaborate more on the issue to find out what can be improved.

Alleviation

The team acknowledged the issue and stated the following:

"In this current version (STFX 1.0) we are forcing managers to reveal most details about their planned trade, to slowly introduce our concept into the market. However, our plan is to gradually allow successful managers to be able to share less and less information about their planned idea in the coming versions. Essentially, a newcomer is required to be more transparent, while a manager that will have gained reputation and/or profits for the community will be allowed to share less and less details in advance. We are currently working on such a "reputation" formula and will be introducing it in Q1 2023."

SVS-09 | LOGICAL ISSUE ABOUT LEVERAGE

Category	Severity	Location	Status
Logical Issue	● Informational	src/StfxVault.sol (8f77134dea22477c18f8ff6f96bf300f6c5f83df): 136	● Acknowledged

Description

The manager can choose leverage to open the position. In the high-leverage position, a margin call is required inevitably. But the protocol seems not to support the operation, since the manager can not flexibly use the funds and he/she can only open one position.

Recommendation

We would like to confirm with the client if the current implementation aligns with the original project design. And whether it is a defect or a feature to restrict managers to open a position once.

Alleviation

The team acknowledged the issue and stated the following:

"Yes. for now, this is the intended design as the manager can only open and manage a single position at a time."

APPENDIX | STFX

Finding Categories

Categories	Description
Centralization / Privilege	Centralization / Privilege findings refer to either feature logic or implementation of components that act against the nature of decentralization, such as explicit ownership or specialized access roles in combination with a mechanism to relocate funds.
Logical Issue	Logical Issue findings detail a fault in the logic of the linked code, such as an incorrect notion on how <code>block.timestamp</code> works.
Control Flow	Control Flow findings concern the access control imposed on functions, such as owner-only functions being invoke-able by anyone under certain circumstances.
Volatile Code	Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that may result in a vulnerability.
Coding Style	Coding Style findings usually do not affect the generated byte-code but rather comment on how to make the codebase more legible and, as a result, easily maintainable.

Checksum Calculation Method

The "Checksum" field in the "Audit Scope" section is calculated as the SHA-256 (Secure Hash Algorithm 2 with digest size of 256 bits) digest of the content of each file hosted in the listed source repository under the specified commit.

The result is hexadecimal encoded and is the same as the output of the Linux "sha256sum" command against the target file.

DISCLAIMER | CERTIK

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to you ("Customer" or the "Company") in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes, nor may copies be delivered to any other person other than the Company, without CertiK's prior written consent in each instance.

This report is not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts CertiK to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. CertiK's position is that each company and individual are responsible for their own due diligence and continuous security. CertiK's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

The assessment services provided by CertiK is subject to dependencies and under continuing development. You agree that your access and/or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon, multiple layers of third-parties.

ALL SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF ARE PROVIDED "AS IS" AND "AS AVAILABLE" AND WITH ALL FAULTS AND DEFECTS WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, CERTIK HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, CERTIK SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AND ALL WARRANTIES ARISING FROM COURSE OF DEALING, USAGE, OR TRADE PRACTICE. WITHOUT LIMITING THE FOREGOING, CERTIK MAKES NO WARRANTY OF ANY KIND THAT THE SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF, WILL MEET CUSTOMER'S OR ANY OTHER PERSON'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULT, BE COMPATIBLE OR WORK WITH ANY SOFTWARE, SYSTEM, OR OTHER SERVICES, OR BE SECURE, ACCURATE, COMPLETE, FREE OF HARMFUL CODE, OR ERROR-FREE. WITHOUT LIMITATION TO THE

FOREGOING, CERTIK PROVIDES NO WARRANTY OR UNDERTAKING, AND MAKES NO REPRESENTATION OF ANY KIND THAT THE SERVICE WILL MEET CUSTOMER'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULTS, BE COMPATIBLE OR WORK WITH ANY OTHER SOFTWARE, APPLICATIONS, SYSTEMS OR SERVICES, OPERATE WITHOUT INTERRUPTION, MEET ANY PERFORMANCE OR RELIABILITY STANDARDS OR BE ERROR FREE OR THAT ANY ERRORS OR DEFECTS CAN OR WILL BE CORRECTED.

WITHOUT LIMITING THE FOREGOING, NEITHER CERTIK NOR ANY OF CERTIK'S AGENTS MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND, EXPRESS OR IMPLIED AS TO THE ACCURACY, RELIABILITY, OR CURRENCY OF ANY INFORMATION OR CONTENT PROVIDED THROUGH THE SERVICE. CERTIK WILL ASSUME NO LIABILITY OR RESPONSIBILITY FOR (I) ANY ERRORS, MISTAKES, OR INACCURACIES OF CONTENT AND MATERIALS OR FOR ANY LOSS OR DAMAGE OF ANY KIND INCURRED AS A RESULT OF THE USE OF ANY CONTENT, OR (II) ANY PERSONAL INJURY OR PROPERTY DAMAGE, OF ANY NATURE WHATSOEVER, RESULTING FROM CUSTOMER'S ACCESS TO OR USE OF THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS.

ALL THIRD-PARTY MATERIALS ARE PROVIDED "AS IS" AND ANY REPRESENTATION OR WARRANTY OF OR CONCERNING ANY THIRD-PARTY MATERIALS IS STRICTLY BETWEEN CUSTOMER AND THE THIRD-PARTY OWNER OR DISTRIBUTOR OF THE THIRD-PARTY MATERIALS.

THE SERVICES, ASSESSMENT REPORT, AND ANY OTHER MATERIALS HEREUNDER ARE SOLELY PROVIDED TO CUSTOMER AND MAY NOT BE RELIED ON BY ANY OTHER PERSON OR FOR ANY PURPOSE NOT SPECIFICALLY IDENTIFIED IN THIS AGREEMENT, NOR MAY COPIES BE DELIVERED TO, ANY OTHER PERSON WITHOUT CERTIK'S PRIOR WRITTEN CONSENT IN EACH INSTANCE.

NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS.

THE REPRESENTATIONS AND WARRANTIES OF CERTIK CONTAINED IN THIS AGREEMENT ARE SOLELY FOR THE BENEFIT OF CUSTOMER. ACCORDINGLY, NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH REPRESENTATIONS AND WARRANTIES AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH REPRESENTATIONS OR WARRANTIES OR ANY MATTER SUBJECT TO OR RESULTING IN INDEMNIFICATION UNDER THIS AGREEMENT OR OTHERWISE.

FOR AVOIDANCE OF DOUBT, THE SERVICES, INCLUDING ANY ASSOCIATED ASSESSMENT REPORTS OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

CertiK | Securing the Web3 World

Founded in 2017 by leading academics in the field of Computer Science from both Yale and Columbia University, CertiK is a leading blockchain security company that serves to verify the security and correctness of smart contracts and blockchain-based protocols. Through the utilization of our world-class technical expertise, alongside our proprietary, innovative tech, we're able to support the success of our clients with best-in-class security, all whilst realizing our overarching vision; provable trust for all throughout all facets of blockchain.

