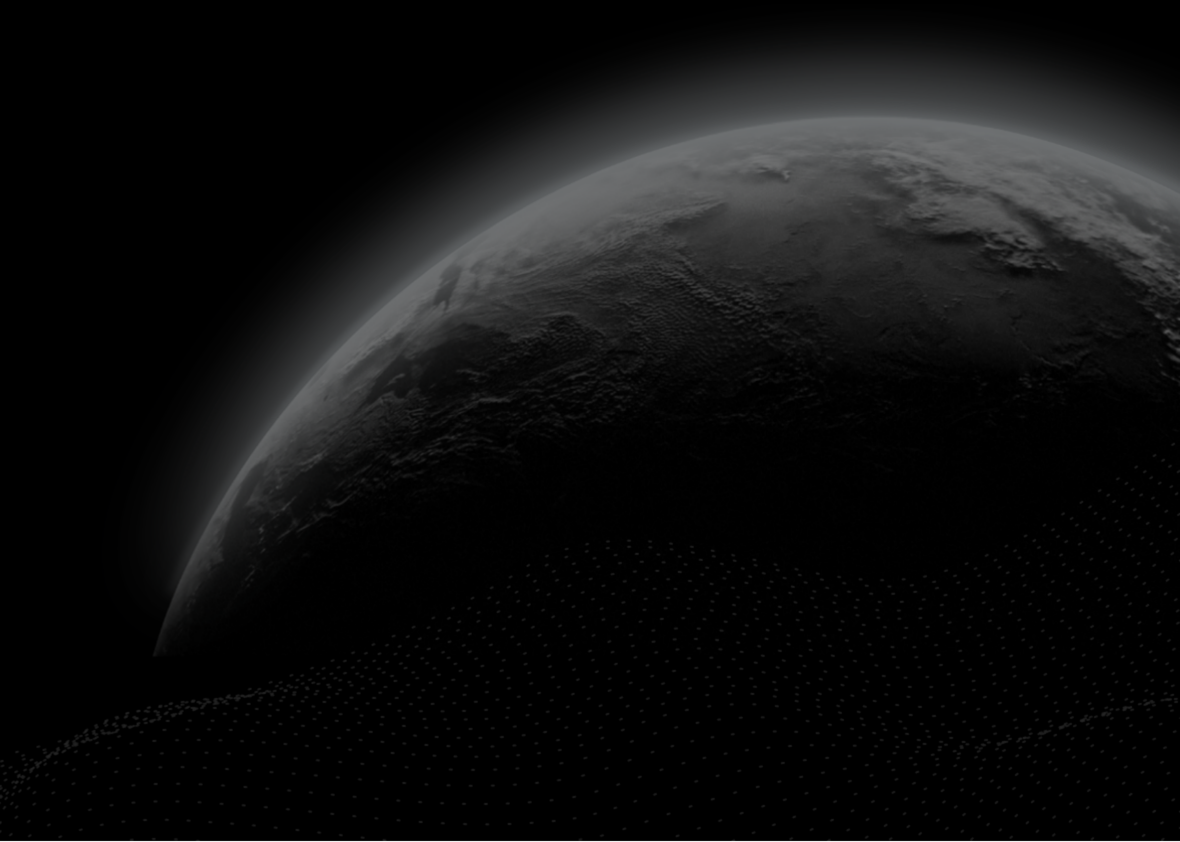
CERTIK

Security Assessment

Baklava Space - Audit

CertiK Assessed on Apr 9th, 2024



CertiK Assessed on Apr 9th, 2024

Baklava Space - Audit

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

Executive Summary

TYPES	ECOSYSTEM	METHODS
DeFi	EVM Compatible	Manual Review, Static Analysis

LANGUAGE	TIMELINE	KEY COMPONENTS
Solidity	Delivered on 04/09/2024	N/A

CODEBASE	COMMITS
stFX-vault View All in Codebase Page	176fee03bdb2f9cff778de1eda748e1280dea069 View All in Codebase Page

Highlighted Centralization Risks

- ⓘ Privileged role can remove users' tokens
- ⓘ Contract upgradeability

Vulnerability Summary

Total Findings	Resolved	Mitigated	Partially Resolved	Acknowledged	Declined
10	4	0	0	6	0

🟥	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.
🟧	2	Major	2 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.
🟨	4	Medium	1 Resolved, 3 Acknowledged	Medium risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform.
🟨	4	Minor	3 Resolved, 1 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.
🟦	0	Informational		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 | BAKLAVA SPACE - AUDIT

CODEBASE | BAKLAVA SPACE - AUDIT

Repository

stFX-vault

Commit

176fee03bdb2f9cff778de1eda748e1280dea069

AUDIT SCOPE | BAKLAVA SPACE - AUDIT

1 file audited ● 1 file without findings

ID	Repo	File	SHA256 Checksum
● SFV	baklavaspace/stFX-vault	StakeFXVaultV2.sol	704a3b8974cc14f6c2ab38610b1bebf723e77e3b866c86a5f31e675573d11430

APPROACH & METHODS | BAKLAVA SPACE - AUDIT

This report has been prepared for Baklava Space to discover issues and vulnerabilities in the source code of the Baklava Space - Audit 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.

FINDINGS | BAKLAVA SPACE - AUDIT

10	0	2	4	4	0
Total Findings	Critical	Major	Medium	Minor	Informational

This report has been prepared to discover issues and vulnerabilities for Baklava Space - Audit. Through this audit, we have uncovered 10 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
SFX-11	**Centralized Control Of Contract Upgrade**	**Centralization**	**Major**	● **Acknowledged**
SFX-12	**Centralization Risks In StakeFXVaultV2.Sol**	**Centralization**	**Major**	● **Acknowledged**
SFX-03	Rewards Not Transferred To User Before Share Transfer	Logical Issue	Medium	● Acknowledged
SFX-04	Out Of Scope Dependencies	Design Issue	Medium	● Acknowledged
SFX-05	Lack Of Storage Gap In Upgradeable Contract	Logical Issue	Medium	● Acknowledged
SFX-13	Potential Underflow Error In `removeValidator`	Logical Issue	Medium	● Resolved
SFX-07	No Upper Limit For Fee	Logical Issue	Minor	● Acknowledged
SFX-08	Potential Miscalculations In `_calculateNumberofValidators`	Logical Issue	Minor	● Resolved
SFX-10	Lack Of Test Coverage	Coding Style	Minor	● Resolved
SFX-15	Incompatibility With Deflationary Tokens	Volatile Code	Minor	● Resolved

SFX-11 | CENTRALIZED CONTROL OF CONTRACT UPGRADE

Category	Severity	Location	Status
Centralization	● **Major**	**StakeFXVaultV2.sol (176fee0): 18**	● **Acknowledged**

Description

In the contract `StakeFXVaultV2`, the role `OWNER_ROLE` has the authority to update the implementation contract.

Any compromise to the `OWNER_ROLE` account may allow a hacker to take advantage of this authority and change the implementation contract which is pointed by proxy and therefore execute potential malicious functionality in the implementation contract.

Recommendation

We recommend that the team make efforts to restrict access to the admin of the proxy contract. A strategy of combining a time-lock and a multi-signature (⅔, ⅗) wallet can be used to prevent a single point of failure due to a private key compromise. In addition, the team should be transparent and notify the community in advance whenever they plan to migrate to a new implementation contract.

Here are some feasible short-term and long-term suggestions that would mitigate the potential risk to a different level and suggestions that would permanently fully resolve the risk.

Short Term:

A combination of a time-lock and a multi signature (⅔, ⅗) wallet mitigate the risk by delaying the sensitive operation and avoiding a single point of key management failure.

- A time-lock with reasonable latency, such as 48 hours, for awareness of privileged operations;
  AND
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to a private key compromised;
  AND
- A medium/blog link for sharing the time-lock contract and multi-signers addresses information with the community.

For remediation and mitigated status, please provide the following information:

- Provide the deployed time-lock address.

- Provide the **gnosis** address with **ALL** the multi-signer addresses for the verification process.

- Provide a link to the **medium/blog** with all of the above information included.

Long Term:

A combination of a time-lock on the contract upgrade operation and a DAO for controlling the upgrade operation mitigate the contract upgrade risk by applying transparency and decentralization.

- A time-lock with reasonable latency, such as 48 hours, for community awareness of privileged operations;
  AND
- Introduction of a DAO, governance, or voting module to increase decentralization, transparency, and user involvement;
  AND
- A medium/blog link for sharing the time-lock contract, multi-signers addresses, and DAO information with the community.

For remediation and mitigated status, please provide the following information:

- Provide the deployed time-lock address.
- Provide the **gnosis** address with **ALL** the multi-signer addresses for the verification process.
- Provide a link to the **medium/blog** with all of the above information included.

Permanent:

Renouncing ownership of the `admin` account or removing the upgrade functionality can *fully* resolve the risk.

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

Note: we recommend the project team consider the long-term solution or the permanent solution. The project team shall make a decision based on the current state of their project, timeline, and project resources.

Alleviation

[Baklava Space Team, 04/02/2024]: Deploy multisig and transfer ownership.

[CertiK, 04/02/2024]: It is suggested to implement the aforementioned methods to avoid centralized failure. Also, CertiK strongly encourages the project team to periodically revisit the private key security management of all addresses related to centralized roles.

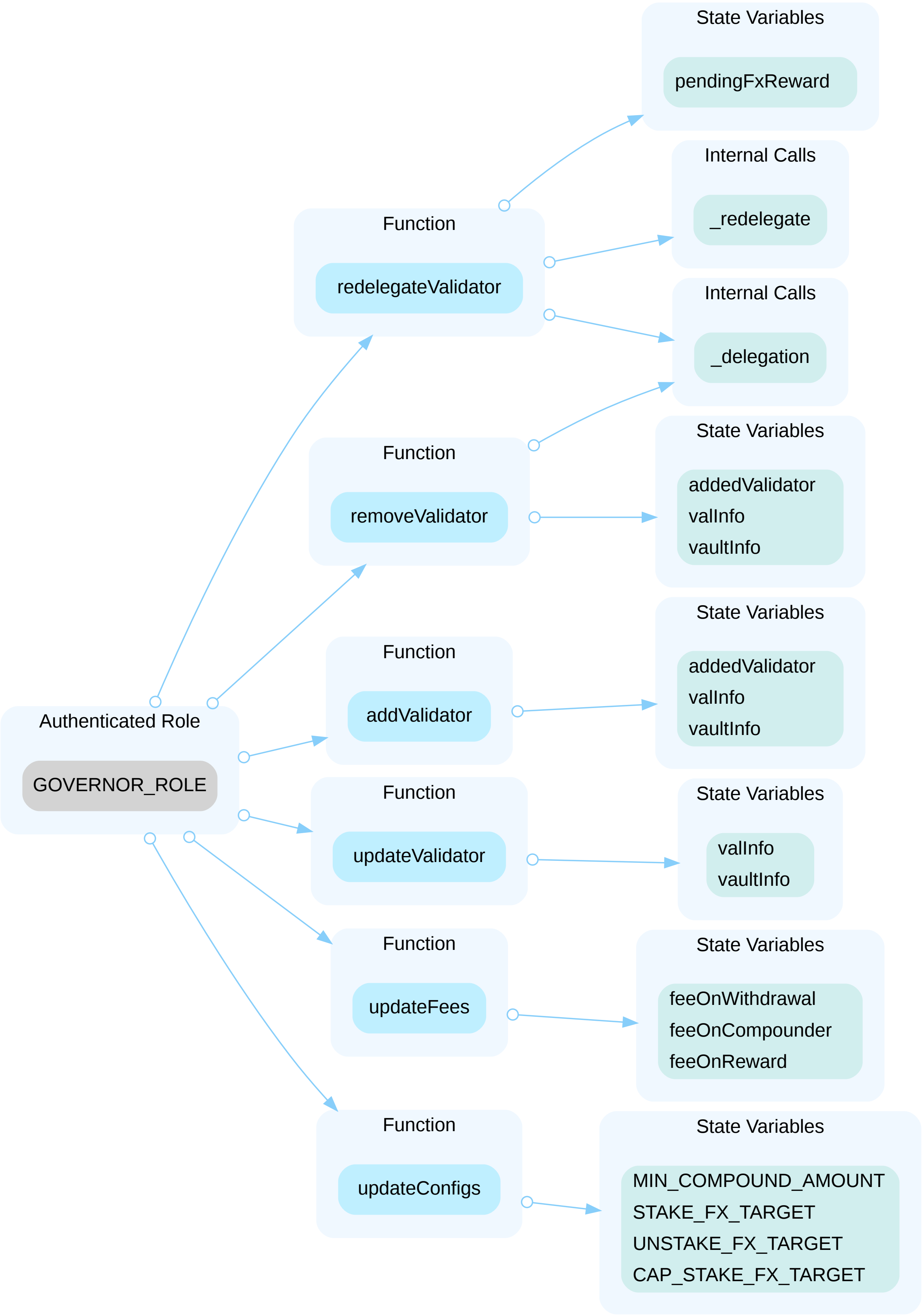
SFX-12 | CENTRALIZATION RISKS IN STAKEFXVAULTV2.SOL

Category	Severity	Location	Status
Centralization	● **Major**	**StakeFXVaultV2.sol (176fee0): 213, 491, 555, 574, 597, 618, 631, 638, 646, 651, 656, 661**	● **Acknowledged**

Description

In the contract `StakeFXVaultV2` the role `GOVERNOR_ROLE` has authority over the functions shown in the diagram below. Any compromise to the `GOVERNOR_ROLE` account may allow the hacker to take advantage of this authority.

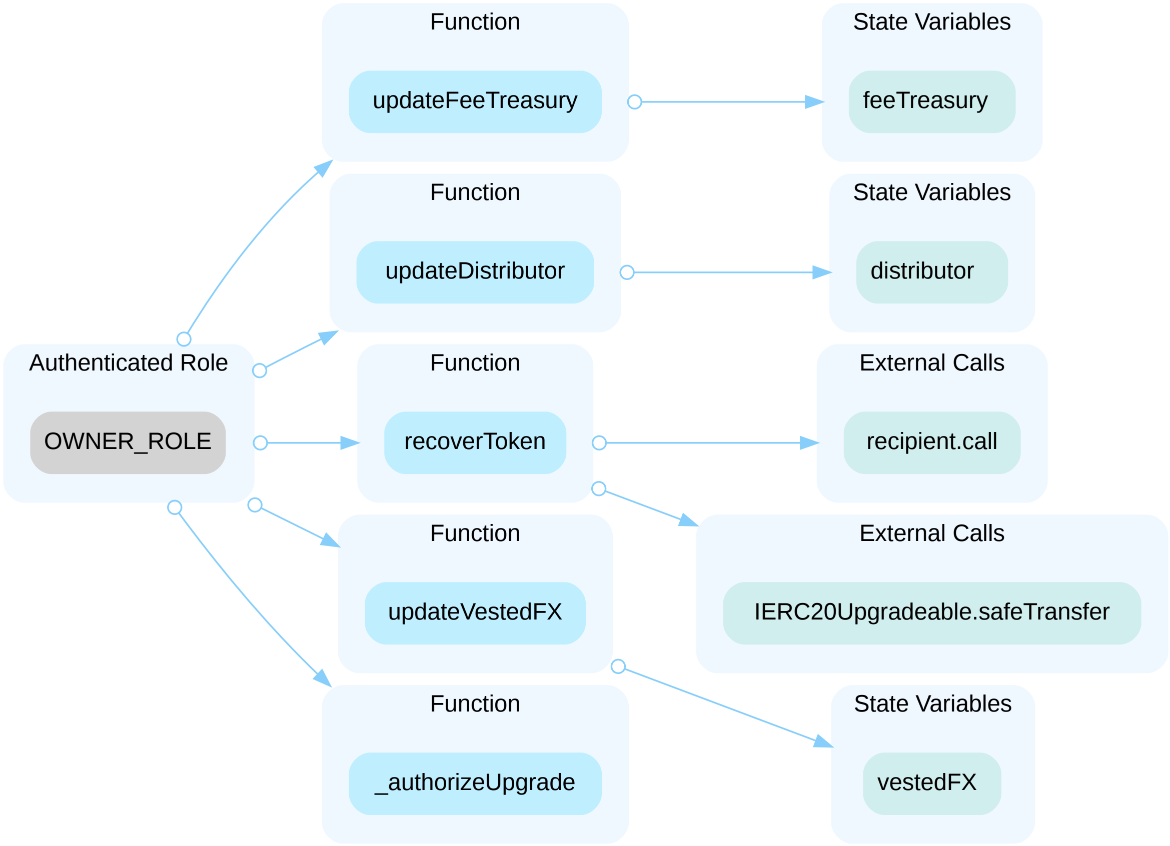
- Add the validator through `addValidator`
- Remove validators which has 0 allocPoint and 0 delegation in the list through `addValidator`
- Transfer share from validator to another validator through `redelegateValidator`
- update `allocPoint` for a validator through `updateValidator`
- update variables `MIN_COMPOUND_AMOUNT` , `CAP_STAKE_FX_TARGET` , `UNSTAKE_FX_TARGET` , `STAKE_FX_TARGET`



In the contract `StakeFXVaultV2` the role `OWNER_ROLE` has authority over the functions shown in the diagram below. Any compromise to the `OWNER_ROLE` account may allow the hacker to take advantage of this authority.

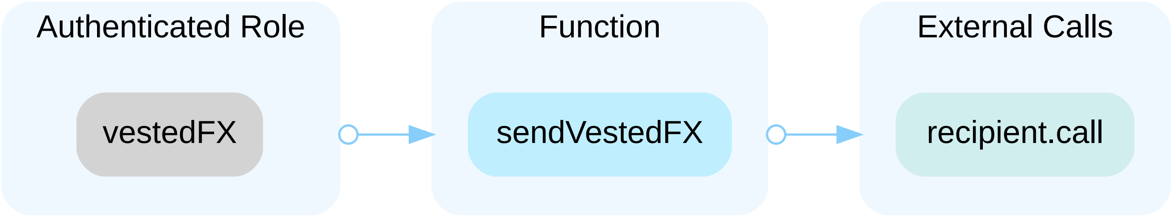
- Upgrade the contract

- Update `vestedFX` address through `updateVestedFX`
- Update `feeTreasury` address through `updateFeeTreasury`
- Update `distributor` address through `updateDistributor`
- withdraw the tokens or FX from the contract through `recoverToken`



In the contract `StakeFXVaultV2` the role `vestedFX` has authority over the functions shown in the diagram below. Any compromise to the `vestedFX` account may allow the hacker to take advantage of this authority.

- withdraw FX from the contract through `sendVestedFX`



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 advise the client to carefully manage 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., multisignature 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 (⅔, ⅗) 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.

Alleviation

[Baklava Space Team, 04/02/2024]: Deploy multisig and transfer ownership.

[CertiK, 04/02/2024]: It is suggested to implement the aforementioned methods to avoid centralized failure. Also, CertiK strongly encourages the project team to periodically revisit the private key security management of all addresses related to centralized roles.

SFX-03 | REWARDS NOT TRANSFERRED TO USER BEFORE SHARE TRANSFER

Category	Severity	Location	Status
Logical Issue	● Medium	StakeFXVaultV2.sol (176fee0): 191	● Acknowledged

Description

The `entrustDelegatedShare()` function is intended to transfer a user's delegated shares to the contract. However, there is a logic problem where rewards generated from user shares are not returned to the user. Instead, the returned rewards are recorded in the variable `pendingFxReward` for compounding.

```
189          (uint256 fxAmountToTransfer, uint256 returnReward) =
_transferFromShares(val, msg.sender, address(this), amount);
190
191          pendingFxReward += returnReward;
```

Recommendation

We recommend that rewards generated from user shares are promptly returned to the respective users.

Alleviation

[Baklava Space Team, 04/02/2024]: The team acknowledged the finding and decided not to change the current codebase. transferFrom reward is "to(contract)" rewards, not user rewards.

SFX-04 | OUT OF SCOPE DEPENDENCIES

Category	Severity	Location	Status
Design Issue	● Medium	StakeFXVaultV2.sol (176fee0): 12, 13, 14, 15, 16	● Acknowledged

Description

The contract `StakeFXVaultV2` import several out-of-scope contracts **IVestedFX**, **IRewardDistributor**, **IWFX**, **BaseVault**, **vaultUtils**,**PrecompileStaking**. The scope of the audit treats out-of-scope contracts as black boxes and assumes their functional correctness. However, in the real world, out-of-scope contracts can be compromised and this may lead to lost or stolen assets.

Recommendation

We recommend that the project team constantly monitor the functionality of the out-of-scope files to mitigate any side effects that may occur when unexpected errors are discovered.

Alleviation

[Baklava Space Team, 04/02/2024]: The team acknowledged the finding and decided not to change the current codebase.

SFX-05 | LACK OF STORAGE GAP IN UPGRADEABLE CONTRACT

Category	Severity	Location	Status
Logical Issue	● Medium	StakeFXVaultV2.sol (176fee0): 18	● Acknowledged

Description

There is no storage gap preserved in the logic contract. Any logic contract that acts as a base contract that needs to be inherited by other upgradeable child should have a reasonable size of storage gap preserved for the new state variable introduced by the future upgrades.

Recommendation

We recommend having a storage gap of a reasonable size preserved in the logic contract in case that new state variables are introduced in future upgrades. For more information, please refer to: https://docs.openzeppelin.com/contracts/3.x/upgradeable#storage_gaps.

Alleviation

[Baklava Space Team, 04/02/2024]: The team acknowledged the finding and decided not to change the current codebase. Storage gap has been preserved in all parent contracts (BaseVault.sol, PrecompileStaking.sol, Governable.sol)

SFX-13 | POTENTIAL UNDERFLOW ERROR IN `removeValidator`

Category	Severity	Location	Status
Logical Issue	● Medium	StakeFXVaultV2.sol (176fee0): 590	● Resolved

Description

The `removeValidator()` function is designed to remove nodes from the list with 0 allocPoint and delegation. If the first node meets the removal conditions, an underflow situation occurs due to the attempt to decrement `i` at line 590 when it is already zero.

```
578         for (uint256 i = 0; i < vaultLength; i++) {
579             if (valInfo[i].allocPoint == 0) {
580                 string memory val = valInfo[i].validator;
581                 (uint256 sharesAmount, ) = _delegation(val, address(this));
582                 if (sharesAmount == 0) {
583                     addedValidator[val] = false;
584                     uint256 lastIndex = vaultLength - 1;
585                     valInfo[i] = valInfo[lastIndex];
586                     delete valInfo[lastIndex];
587 
588                     emit ValidatorRemoved(val);
589                     vaultLength--;
590                     i--;
591                 }
592             }
593         }
```

Recommendation

Modify the `removeValidator` function to handle the decrement of `i` when the first node satisfies the removal conditions, preventing the occurrence of underflow.

Alleviation

[Baklava Space Team, 04/02/2024]: The team heeded the advice and resolved the issue in commit: d78044529d2c130f62d24778d3b9e25b398c23f1.

SFX-07 | NO UPPER LIMIT FOR FEE

Category	Severity	Location	Status
Logical Issue	● Minor	StakeFXVaultV2.sol (176fee0): 639, 640, 641	● Acknowledged

Description

There are no upper boundaries for `feeOnReward` , `feeOnCompounder` , `feeOnWithdrawal` which are used to calculate fee for protocol, and charge user for staking. It is possible to set the total fee rate up to any arbitrary amount.

Recommendation

Introduce a maximum fee threshold in the function to ensure fee values remain within acceptable limits. This safeguard will provide predictability and fairness in fee-related operations.

Alleviation

[Baklava Space Team, 04/02/2024]: we have added notice in NatSpec comment to remind developer/owner to take note when calling this function.

SFX-08 | POTENTIAL MISCALCULATIONS IN `_calculateNumberofValidators`

Category	Severity	Location	Status
Logical Issue	● Minor	StakeFXVaultV2.sol (176fee0): 416	● Resolved

Description

The function `_calculateNumberofValidators()` is designed to compute the number of validators required for a given amount of tokens. The line 415 checks if the amount without precision is greater than 10, indicating that each validator should deposit 10 * 10^18 tokens each time, line 416 mistakenly divides the `delegateAmountInEther` by 10 instead of subtracting 10.

```
415         while (delegateAmountInEther >= 10) {
416             delegateAmountInEther /= 10;
417             numValidators++;
418         }
```

Recommendation

Adjust line 416 to subtract 10 from `delegateAmountInEther` instead of dividing by 10.

Alleviation

[Baklava Space Team, 04/02/2024]: logbase10 logic to calc number of validators.

SFX-10 | LACK OF TEST COVERAGE

Category	Severity	Location	Status
Coding Style	● Minor	StakeFXVaultV2.sol (176fee0): 18	● Resolved

Description

No unit tests for StakeFXVaultV2.sol were provided. Testing programs are a very important aspect of proving program correctness, preventing regressions, and release engineering. Without tests, there is way more difficult to know if the program works as expected.

Recommendation

We recommend writing tests for StakeFXVaultV2.sol.

Alleviation

[Baklava Space Team, 04/02/2024]: The team heeded the advice and resolved the issue in commit: d78044529d2c130f62d24778d3b9e25b398c23f1.

SFX-15 | INCOMPATIBILITY WITH DEFLATIONARY TOKENS

Category	Severity	Location	Status
Volatile Code	● Minor	StakeFXVaultV2.sol (176fee0): 129	● Resolved

Description

When transferring standard ERC20 deflationary tokens, the input amount may not be equal to the received amount due to the charged transaction fee. As a result, an inconsistency in the amount will occur and the transaction may fail due to the validation checks. For example, if a user sends 100 deflationary tokens (with a 10% transaction fee) to the target contract, only 90 tokens actually arrive to the contract.

Recommendation

We recommend regulating the set of LP tokens supported and adding necessary mitigation mechanisms to keep track of accurate balances if there is a need to support deflationary tokens.

Alleviation

[Baklava Space Team, 04/02/2024]: WFX is not deflationary tokens (1WFX = 1 FX).

APPENDIX | BAKLAVA SPACE - AUDIT

Finding Categories

Categories	Description
Coding Style	Coding Style findings may not affect code behavior, but indicate areas where coding practices can be improved to make the code more understandable and maintainable.
Volatile Code	Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases and may result in vulnerabilities.
Logical Issue	Logical Issue findings indicate general implementation issues related to the program logic.
Centralization	Centralization findings detail the design choices of designating privileged roles or other centralized controls over the code.
Design Issue	Design Issue findings indicate general issues at the design level beyond program logic that are not covered by other finding categories.

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.