CERTIK

Security Assessment

Nutbox Walnut Network

Oct 29th, 2021

Table of Contents

Summary

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

- 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.

Overview

Project Summary

Project Name	Nutbox Walnut Network
Platform	Ethereum
Language	Solidity
Codebase	https://github.com/nutbox-dao/nutbox-contract/tree/release-v1.0
Commit	5b697c23473e1a6eb8face5c77c40dc6d46f6226 0b3dab054fd74f623220960f5fe277ed1969bfe3

Audit Summary

Delivery Date	Oct 29, 2021
Audit Methodology	Static Analysis, Manual Review
Key Components	

Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Partially Resolved	Resolved
Critical	1	0	0	0	0	1
Major	6	0	0	4	0	2
Medium	0	0	0	0	0	0
Minor	6	0	0	4	1	1
Informational	21	0	0	11	3	7
Discussion	0	0	0	0	0	0

Audit Scope

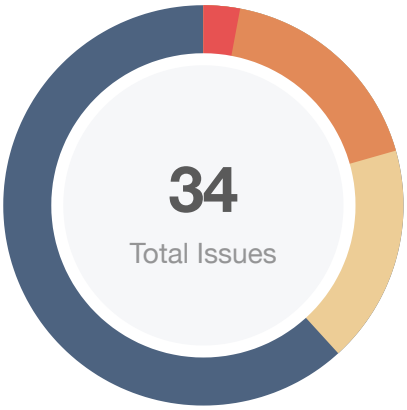
ID	File	SHA256 Checksum
ERC	asset/handler/ERC20AssetHandler.sol	77758f556ae57f798ee6bc7bff0825c39c4ab503a94fd505ad58ac2017a11f8c
ERA	asset/handler/ERC721AssetHandler.sol	6f7619ddf858a22775315f76d4bd65a6bb34e409dbc66c20d61740906982f992
TAH	asset/handler/TrustlessAssetHandler.sol	2bfd19c3ec35921adc5e4a7fdba5cf3c4becfa167bd0858f81fb1de6cdd22655
IAR	asset/interfaces/IAssetRegistry.sol	e79d25542f71493cf2a083ce4143bbbcfca0ac8d47b0bc642e085791ac86868e
IRH	asset/interfaces/IRegistryHub.sol	9995039242baeef53f645be3bb8342d918398c1e6d05975454f5f960a46860dc
ITA	asset/interfaces/ITrustAssetHandler.sol	85a4725936c15f95905cc0c36dc6f6f74cbc24c1aadef9825f7502655c96c233
ITH	asset/interfaces/ITrustlessAssetHandler.sol	53c456af882c44c4bbeff07b0bbcc68b38ea0b39eeca656cc91545712d994a35
HCA	asset/registry/HomeChainAssetRegistry.sol	3fa0136755eb7bda24837b484d5c463456d46877b03e34d33394a25af6ce383a
RHN	asset/registry/RegistryHub.sol	1c43301a3f6bee9acf5e16ad9c8bafab6c3ec845ab0b5828477cecfab588ab0d
SHD	asset/registry/SteemHiveDelegateAssetRegistry.sol	0b7aef5d8fb1a9fa52c3da5856f224c8250b883d61599352806525c9827f1a27
SCA	asset/registry/SubstrateCrowdloanAssetRegistry.sol	2e63dc875846c3b9ba6f0d64f52a95ce1b308b1f5b9b5ddfe8de0493e1acbc6a
SNA	asset/registry/SubstrateNominateAssetRegistry.sol	be577a9dc906f4970a26865c19024681a2f8929ebef47b7a37623802c2bb5b75
ERF	asset/ERC20Factory.sol	0329ba084fcd5a197d7903f565901d5a63dbd1de0ededf06c2122db6cde77c1e
ERH	asset/ERC20Helper.sol	173bcaa31535c77db915c6fa79559e2c2280519f0c99b7e2ae5fd57bc7c9acb9

ID	File	SHA256 Checksum
IBN	bridge/interfaces/IBridge.sol	8a0e3d7d64340b8ce24a47c10f4b17ef39690a1369ddc7eba0f270bf0a304193
IEN	bridge/interfaces/IExecutor.sol	819f57be0faafa111f6ad0c9c89076b8f0d615176b8dfbfc8e2776a1b8dd36a9
BNC	bridge/Bridge.sol	26b30974a3938d28531950414141f3cd3ea28a5d3018accc9140bd31a68238aa
ENC	bridge/Executor.sol	88a59bef09d583d1310c0cc206423527cd9871400758f228820d023816ed5122
BLN	common/libraries/BytesLib.sol	368d6fdfe026f071c971732da1abcc1fbdd6c6bda573967d9dfce477c5b8f98f
TNC	common/Types.sol	e7b7ccf2b519eab5f9459bea35851d9b8ab287839d555ae5bbaf05d5f5ea524c
ICN	staking/calculators/ICalculator.sol	f36c2bd487980d638fe2ea397b5bd6fe84e6ba5fdc185ab684b412463fc46b06
LCN	staking/calculators/LinearCalculator.sol	cbb7771a0164d2d0ae51c5759838681879da17bd94b6ef7f8d922f8bd1625b32
SFN	staking/StakingFactory.sol	f1bf400d7215b76c01cea37f923b72ee1a6486e04d917026ab703a328ffbc703
STN	staking/StakingTemplate.sol	4b5d37e27ef3be4a3dee738e1f70d0796f74796e72ccded2fbe2cf30266a969f
MER	MintableERC20.sol	b196914084897a28a3cf865cea10f7b82aa242f5cb164646ede1c3b14fb3dbf6
NUT	NUTToken.sol	eef6e3baa2f7d91c99773d2a0a10328b7b9d8ba3c292e741a56e08ef85c41bd2
NDC	NoDelegateCall.sol	2787b95a9871f8de1c20eb3a8cc7172b64ddc34cfa81cef97a55d2757b708ccb
SER	SimpleERC20.sol	d16172305f8f6bc6382c053887c0dda6bffb382bc65f663b3ab499accaf80d39

It should be noted that the system design includes a number of economic arguments and assumptions. These were explored to the extent that they clarified the intention of the code base, but we did not audit the mechanism design itself.

Additionally, financial models of blockchain protocols need to be resilient to attacks. It needs to pass simulations and verifications to guarantee the security of the overall protocol. The accuracy of the financial model is not in the scope of the audit.

Findings



Critical	**1** (2.94%)
Major	**6** (17.65%)
Medium	**0** (0.00%)
Minor	**6** (17.65%)
Informational	**21** (61.76%)
Discussion	**0** (0.00%)

ID	Title	Category	Severity	Status
GLOBAL-01	Unlocked Compiler Version	Language Specific	Informational	Acknowledged
GLOBAL-02	Visibility Specifiers Missing	Language Specific	Informational	Acknowledged
GLOBAL-03	Address Type Could Be Indexed In Events	Gas Optimization	Informational	Partially Resolved
GLOBAL-04	Initial Token Distribution	**Centralization / Privilege**	**Major**	Acknowledged
GLOBAL-05	Centralization Risk	**Centralization / Privilege**	**Major**	Acknowledged
GLOBAL-06	Lack of Zero Address Validation	Volatile Code	Minor	Partially Resolved
GLOBAL-07	Comparison to A Boolean Constant	Optimization	Informational	Acknowledged
GLOBAL-08	Missing Emit Events	Optimization	Informational	Partially Resolved
GLOBAL-09	Unused Variable/Struct	Logical Issue	Informational	Partially Resolved
GLOBAL-10	Set `constant` to Variables	Gas Optimization	Informational	Acknowledged
GLOBAL-11	No Functions In The Contract `ERC721AssetHandler`	Control Flow	Informational	Acknowledged
GLOBAL-12	Set `immutable` to Variables	Gas Optimization	Informational	Resolved
GLOBAL-13	Function Visibility Optimization	Gas Optimization	Informational	Resolved
GLOBAL-14	Incompatibility With Deflationary Tokens	Logical Issue	Minor	Acknowledged

ID	Title	Category	Severity	Status
GLOBAL-15	Lack of Access Control	Logical Issue	Informational	Acknowledged
GLOBAL-16	Repeated Authentication	Control Flow	Informational	Acknowledged
GLOBAL-17	Logic Issue Of Function `setWhitelist`	Logical Issue	Informational	Resolved
GLOBAL-18	Discussion For Function `addPool()`	Logical Issue	Informational	Acknowledged
BNC-01	Not Update `relayerCount`	Logical Issue	Major	Resolved
BNC-02	Logic Issue Of Function `cancelProposal`	Logical Issue	Minor	Resolved
BNC-03	Redundant Code	Logical Issue	Informational	Resolved
ENC-01	Logic Issue Of Function `executeProposal`	Logical Issue	Informational	Acknowledged
ERH-01	Unrestricted Access on `fundERC20`	Logical Issue	Major	Resolved
NDC-01	Lack of Error Message	Coding Style	Informational	Acknowledged
NUT-01	Visibility Specifiers Missing	Language Specific	Informational	Resolved
NUT-02	Address Type Could Be Indexed In Events	Gas Optimization	Informational	Resolved
NUT-03	Initial Token Distribution	**Centralization / Privilege**	**Major**	Acknowledged
NUT-04	Redundant Code	Logical Issue	Informational	Resolved
NUT-05	Centralization Risk	**Centralization / Privilege**	**Major**	Acknowledged
STN-01	Check Effect Interaction Pattern Violated	Logical Issue	Minor	Acknowledged
STN-02	Lack of Pool Validity Checks	Logical Issue	Informational	Acknowledged
STN-03	Add `require` In Function `tryWithdraw`	Logical Issue	Minor	Acknowledged
STN-04	Logic Issue Of Function `adminWithdrawReward`	Logical Issue	Critical	Resolved

ID	Title	Category	Severity	Status
STN-05	Add `require` In Function `removePool`	Logical Issue	Minor	Acknowledged

GLOBAL-01 | Unlocked Compiler Version

Category	Severity	Location	Status
Language Specific	Informational	Global	Acknowledged

Description

The contract contains unlocked compiler versions. An unlocked compiler version in the contract's source code permits the user to compile it at or above a particular version. This, in turn, leads to differences in the generated bytecode between compilations due to differing compiler version numbers. This can lead to ambiguity when debugging as compiler-specific bugs may occur in the codebase that would be difficult to identify over a span of multiple compiler versions rather than a specific one.

Recommendation

It is general practice to alternatively lock the compiler at a specific version rather than allow a range of compiler versions to be utilized to avoid compiler-specific bugs and thus be able to detect emerging ones. We recommend locking the compiler at the lowest possible version that supports all the capabilities required by the codebase. This will ensure that the project utilizes a compiler version that has been in use for the longest time and as such is less likely to contain yet-undiscovered bugs.

Alleviation

No alleviation.

GLOBAL-02 | Visibility Specifiers Missing

Category	Severity	Location	Status
Language Specific	Informational	Global	Acknowledged

Description

The variable declaration does not have a visibility specifier explicitly set.

Recommendation

Inconsistencies in the default visibility the Solidity compilers impose can cause issues in the functionality of the codebase. We advise that visibility specifier for the linked variable is explicitly set.

Alleviation

No alleviation.

GLOBAL-03 | Address Type Could Be Indexed In Events

Category	Severity	Location	Status
Gas Optimization	Informational	Global	Partially Resolved

Description

It is recommended to add `indexed` keyword for parameters in events, which makes it easier for users to navigate event logs.

Recommendation

We advise the client to add keyword `indexed` in the declaration of events.

Alleviation

The client heeded our advice and partially resolved this issue in commit : 873aa232e1c5c23ae0453967f4109ba661fd5e7b.

GLOBAL-04 | Initial Token Distribution

Category	Severity	Location	Status
Centralization / Privilege	**Major**	**Global**	Acknowledged

Description

`initialSupply` tokens were sent to the `owner` when deploying the contract. This could be a centralization risk as the deployer can distribute tokens without obtaining the consensus of the community.

- `MintableERC20.constructor()`
- `SimpleERC20.constructor()`

Recommendation

We recommend the team to be transparent regarding the initial token distribution process.

Alleviation

No alleviation.

GLOBAL-05 | Centralization Risk

Category	Severity	Location	Status
Centralization / Privilege	Major	Global	Acknowledged

Description

To bridge the gap in trust between the administrators need to express a sincere attitude regarding the considerations of the administrator team's anonymity.

The `MINTER_ROLE` of `MintableERC20` has the responsibility to notify users about the following capabilities:

- mint uncapped tokens to any address through `mint()`

The `owner` of `RegistryHub` has the responsibility to notify users about the following capabilities:

- set asset handlers through `setAssetHandlers()`
- set whitelist through `setWhiteList()`
- set `NUT` and `stakedNUT` through `setNUTStaking()`

The `whiteList` of `RegistryHub` has the responsibility to notify users about the following capabilities:

- register through `add()`
- set mintable through `setMintable()`

The `owner` of `HomeChainAssetRegistry/SteemHiveDelegateAssetRegistry/SubstrateCrowdloanAssetRegistry/SubstrateNominateAssetRegistry` has the responsibility to notify users about the following capabilities:

- set the address of `registryHub` through `setRegistryHub()`

The `DEFAULT_ADMIN_ROLE` of `ERC20AssetHandler` has the responsibility to notify users about the following capabilities:

- set the address of `registryHub` through `setRegistryHub()`
- grant `WHITELIST_MANAGER_ROLE` role to `_manager` through `adminAddWhitelistManager()`
- revoke `WHITELIST_MANAGER_ROLE` role from `_manager` through `adminRemoveWhitelistManager()`
- set whitelist through `setWhiteList()`

The `WHITELIST_MANAGER_ROLE` of `ERC20AssetHandler` has the responsibility to notify users about the following capabilities:

- set whitelist through `setWhiteList()`

The `whiteList` of `ERC20AssetHandler` has the responsibility to notify users about the following capabilities:

- lock or burn asset through `lockOrBurnAsset()`
- lock asset through `lockAsset()`
- unlock or mint asset through `unlockOrMintAsset()`
- unlock asset through `unlockAsset()`

The `DEFAULT_ADMIN_ROLE` of `TrustlessAssetHandler` has the responsibility to notify users about the following capabilities:

- set the address of `executor` through `setExecutor()`
- set the address of `registryHub` through `setRegistryHub()`
- grant `WHITELIST_MANAGER_ROLE` role to `_manager` through `adminAddWhitelistManager()`
- revoke `WHITELIST_MANAGER_ROLE` role from `_manager` through `adminRemoveWhitelistManager()`
- set whitelist through `setWhiteList()`

The `WHITELIST_MANAGER_ROLE` of `TrustlessAssetHandler` has the responsibility to notify users about the following capabilities:

- set whitelist through `setWhiteList()`

The `whiteList` of `TrustlessAssetHandler` has the responsibility to notify users about the following capabilities:

- attach pool through `attachPool()`

The `executor` of `TrustlessAssetHandler` has the responsibility to notify users about the following capabilities:

- update balance through `updateBalance()`

The `DEFAULT_ADMIN_ROLE` of `Bridge` has the responsibility to notify users about the following capabilities:

- set the address of `executor` through `adminSetExecutor()`
- add relayer through `adminAddRelayer()`
- remove relayer through `adminRemoveRelayer()`
- set `threshold` through `adminSetThreshold()`
- set `fee` through `adminSetFee()`

- set `expiry` through `adminSetExpiry()`
- grant `DEFAULT_ADMIN_ROLE` role to `_newAdmin` and revoke `DEFAULT_ADMIN_ROLE` role from `msg.sender` though `adminRenonceAdmin()`
- deposit asset through `adminDepositAsset()`

The `relayer` of `Bridge` has the responsibility to notify users about the following capabilities:

- vote proposal through `voteProposal()`
- cancel proposal through `cancelProposal()`

The `DEFAULT_ADMIN_ROLE` of `Executor` has the responsibility to notify users about the following capabilities:

- set the address of `bridge` through `adminSetBridge()`
- grant `DEFAULT_ADMIN_ROLE` role to `_newAdmin` and revoke `DEFAULT_ADMIN_ROLE` role from `msg.sender` through `adminRenonceAdmin()`
- execute proposal through `adminExecuteProposal()`

The `bridge` of `Executor` has the responsibility to notify users about the following capabilities:

- execute proposal through `executeProposal()`

The `admin` of `LinearCalculator` has the responsibility to notify users about the following capabilities:

- set the address of `factory` through `adminSetStakingFactory()`

The `factory` of `LinearCalculator` has the responsibility to notify users about the following capabilities:

- set distribution era through `setDistributionEra()`

The `admin` of `StakingTemplate` has the responsibility to notify users about the following capabilities:

- deposit reward though `adminDepositReward()`
- withdraw reward through `adminWithdrawReward()`
- add pool through `addPool()`
- remove pool through `removePool()`
- refund all users through `tryWithdraw()`
- stop pool through `stopPool()`
- set pool ratio through `setPoolRatios()`
- set `admin` through `setAdmin()`
- set `dev` through `setDev()`

- set reward ratio through `setDevRewardRatio()`

The `factory` of `StakingTemplate` has the responsibility to notify users about the following capabilities:

- create staking template contract through `initialize()`

Recommendation

We advise the client to carefully manage the privileged account's private keys to avoid any potential risks of being hacked. In general, we strongly recommend centralized privileges or roles in the protocol to be improved via a decentralized mechanism or via smart-contract-based accounts with enhanced security practices, e.g. Multisignature wallets.

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

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key;
- Introduction of a DAO/governance/voting module to increase transparency and user involvement.

Alleviation

No alleviation.

GLOBAL-06 | Lack of Zero Address Validation

Category	Severity	Location	Status
Volatile Code	Minor	Global	Partially Resolved

Description

The given input is missing the check for the non-zero address. For example:

- contract `ERC20Factory`: `_registryHub` in function `constructor()`
- contract `RegistryHub`: `_erc20AssetHandler`, `_erc721AssetHandler`, `_erc1155AssetHandler` and `_trustlessAssetHandler` in function `constructor()`, `owner` in function `add()`
- contract `HomeChainAssetRegistry`: `_erc20Factory` in function `constructor()`
- contract `LinearCalculator`: `_factory` in function `adminSetStakingFactory()`
- contract `StakingFactory`: `_registryHub` in function `constructor()`, `_rewardCalculator` in function `createStakingFeast()`
- contract `StakingTemplate`: `_admin` in function `setAdmin()`, `_dev` in function `setDev()`

Recommendation

We advise the client to add the check for the passed-in values to prevent unexpected error.

Alleviation

The client heeded our advice and partially resolved this issue in commit : 873aa232e1c5c23ae0453967f4109ba661fd5e7b. The `StakingTemplate` and `StakingFactory` contracts are still.

GLOBAL-07 | Comparison to A Boolean Constant

Category	Severity	Location	Status
Optimization	Informational	Global	Acknowledged

Description

Comparison to a boolean constant. For example:

- contract `RegistryHub`: `#L62`
- contract `Bridge`:`#L60 #L65 #L108 #L114 #L148`

Recommendation

We advise the client to remove the comparison to the boolean constant.

Alleviation

No alleviation.

GLOBAL-08 | Missing Emit Events

Category	Severity	Location	Status
Optimization	Informational	Global	Partially Resolved

Description

Functions that affect the status of sensitive variables should be able to emit events as notifications to customers.

- `RegistryHub.setAssetHandlers()`
- `RegistryHub.setWhiteList()`
- `RegistryHub.setNUTStaking()`
- `RegistryHub.setMintable()`
- `HomeChainAssetRegistry.setRegistryHub()`
- `SteemHiveDelegateAssetRegistry.setRegistryHub()`
- `SubstrateCrowdloanAssetRegistry.setRegistryHub()`
- `SubstrateNominateAssetRegistry.setRegistryHub()`
- `ERC20AssetHandler.setRegistryHub()`
- `ERC20AssetHandler.setWhitelist()`
- `TrustlessAssetHandler.setExecutor()`
- `TrustlessAssetHandler.setRegistryHub()`
- `TrustlessAssetHandler.setWhitelist()`
- `Bridge.adminSetExecutor()`
- `Bridge.adminAddRelayer()`
- `Bridge.adminRemoveRelayer()`
- `Bridge.adminSetThreshold()`
- `Bridge.adminSetFee()`
- `Bridge.adminSetExpiry()`
- `Executor.adminSetBridge()`
- `LinearCalculator.adminSetStakingFactory()`
- `StakingTemplate.setAdmin()`
- `StakingTemplate.setDev()`
- `StakingTemplate.setDevRewardRatio()`

Recommendation

We advise the client to add events for sensitive actions and emit them.

Alleviation

The client heeded our advice and partially resolved this issue in commit : 873aa232e1c5c23ae0453967f4109ba661fd5e7b.

GLOBAL-09 | Unused Variable/Struct

Category	Severity	Location	Status
Logical Issue	Informational	Global	Partially Resolved

Description

The variables/struct are not used in the contract.

- `SteemHiveDelegateAssetRegistry.Properties`
- `SubstrateCrowdloanAssetRegistry.Properties`
- `SubstrateNominateAssetRegistry.Properties`
- `HomeChainAssetRegistry.erc20Factory`
- `Bridge.chainSequence`
- `LinearCalculator.MAX_DISTRIBUTIONS`

Recommendation

We advise the client to remove them if there is no plan for further usage.

Alleviation

The client heeded our advice and partially resolved this issue in commit : 873aa232e1c5c23ae0453967f4109ba661fd5e7b. The struct `Properties` is still not be removed.

GLOBAL-10 | Set `constant` to Variables

Category	Severity	Location	Status
Gas Optimization	Informational	Global	Acknowledged

Description

The variables are unchanged throughout the contract.

- `Executor.version`

Recommendation

We advise the client to set aforementioned variables as `constant` variables.

Alleviation

`[Nutbox]`: We designed this contract can be upgraded. Once we upgrade the contract, we need this variable to tag the version of this contract.

GLOBAL-11 | No Functions In The Contract `ERC721AssetHandler`

Category	Severity	Location	Status
Control Flow	● Informational	Global	ⓘ Acknowledged

Description

The contract `ERC721AssetHandler` is empty. However, other contracts had called some functions. For example:

1. Calling `unlockOrMintAsset` in the contract `Executor`
2. Calling `setWhitelist` in the contract `StakingFactory`

Recommendation

We advise the client to recheck these issues.

Alleviation

`[Nutbox]`: No use of this contract now. We will provide this function in future.

GLOBAL-12 | Set `immutable` to Variables

Category	Severity	Location	Status
Gas Optimization	Informational	Global	Resolved

Description

The variables are only changed once in the `constructor` function. For example:

- `ERC20Factory.registryHub`
- `HomeChainAssetRegistry.erc20Factory`
- `Bridge.registryHub`
- `Executor.registryHub`
- `LinearCalculator.admin`
- `StakingFactory.registryHub`
- `StakingTemplate.registryHub`
- `StakingTemplate.factory`

Recommendation

We advise the client to set aforementioned variables as `immutable` variables.

Alleviation

The client heeded our advice and resolved this issue in commit : 873aa232e1c5c23ae0453967f4109ba661fd5e7b.

GLOBAL-13 | Function Visibility Optimization

Category	Severity	Location	Status
Gas Optimization	Informational	Global	Resolved

Description

`public` functions that are never called by the contract could be declared `external`. When the inputs are arrays, `external` functions are more efficient than `public` functions.

For example:

- `MintableERC20.mint()`
- `ERC20Helper.fundERC20()`
- `ERC20Factory.createERC20()`
- `ERC20AssetHandler.adminAddWhitelistManager()`
- `ERC20AssetHandler.adminRemoveWhitelistManager()`
- `ERC20AssetHandler.setRegistryHub()`
- `ERC20AssetHandler.setWhitelist()`
- `TrustlessAssetHandler.adminAddWhitelistManager()`
- `TrustlessAssetHandler.adminRemoveWhitelistManager()`
- `TrustlessAssetHandler.setRegistryHub()`
- `TrustlessAssetHandler.setWhitelist()`
- `TrustlessAssetHandler.setExecutor()`
- `HomeChainAssetRegistry.setRegistryHub()`
- `RegistryHub.setAssetHandlers()`
- `RegistryHub.setWhiteList()`
- `RegistryHub.setAssetHandlers()`
- `SteemHiveDelegateAssetRegistry.setRegistryHub()`
- `SubstrateCrowdloanAssetRegistry.setRegistryHub()`
- `SubstrateNominateAssetRegistry.setRegistryHub()`
- `Bridge.adminDepositAsset()`
- `Executor.adminSetBridge()`
- `LinearCalculator.adminSetStakingFactory()`
- `LinearCalculator.setDistributionEra()`
- `LinearCalculator.calculateReward()`
- `LinearCalculator.getCurrentRewardPerBlock()`

- `StakingFactory.createStakingFeast()`
- `StakingTemplate.initialize()`
- `StakingTemplate.adminDepositReward()`
- `StakingTemplate.adminWithdrawReward()`
- `StakingTemplate.addPool()`
- `StakingTemplate.removePool()`
- `StakingTemplate.tryWithdraw()`
- `StakingTemplate.stopPool()`
- `StakingTemplate.update()`
- `StakingTemplate.withdrawPoolRewards()`
- `StakingTemplate.withdrawTotalRewards()`
- `StakingTemplate.getUserTotalPendingRewards()`
- `StakingTemplate.getUserStakedAmount()`
- `StakingTemplate.setAdmin()`
- `StakingTemplate.getAdmin()`
- `StakingTemplate.setDev()`
- `StakingTemplate.getDev()`
- `StakingTemplate.setDevRewardRatio()`
- `StakingTemplate.getDevRewardRatio()`
- `StakingTemplate.getUserDepositInfo()`

Recommendation

We advise that the functions' visibility specifiers are set to `external` and the array-based arguments change their data location from `memory` to `calldata`, optimizing the gas cost of the function.

Alleviation

The client heeded our advice and resolved this issue in commit : 873aa232e1c5c23ae0453967f4109ba661fd5e7b.

GLOBAL-14 | Incompatibility With Deflationary Tokens

Category	Severity	Location	Status
Logical Issue	Minor	Global	Acknowledged

Description

The contract operates as the main entry for interaction with staking users. The staking users deposit LP tokens into the pool and in return get a proportionate share of the pool's rewards. Later on, the staking users can withdraw their own assets from the pool. In this procedure, `deposit()` and `withdraw()` are involved in transferring users' assets into (or out of) the protocol. 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 advise the client to regulate the set of LP tokens supported in the contract. If there is a need to support deflationary tokens, add necessary mitigation mechanisms to keep track of accurate balances.

Alleviation

No alleviation.

GLOBAL-15 | Lack of Access Control

Category	Severity	Location	Status
Logical Issue	Informational	Global	Acknowledged

Description

The following functions have no access control, anyone can invoke them. For example:

- `StakingFactory.createStakingFeast()`
- `HomeChainAssetRegistry.registerAsset()`
- `SteemHiveDelegateAssetRegistry.registerAsset()`
- `SubstrateCrowdloanAssetRegistry.registerAsset()`
- `SubstrateNominateAssetRegistry.registerAsset()`

Recommendation

We advise the client to recheck them.

Alleviation

`[Nutbox]`: These contracts are open to everyone. Everyone can create `stakingfeast` and register assets.

GLOBAL-16 | Repeated Authentication

Category	Severity	Location	Status
Control Flow	Informational	Global	Acknowledged

Description

The modifier `AccessControl.onlyRole` and `onlyAdmin` are both that check an account has a specific role.

- `Bridge.adminRenonceAdmin()`
- `Executor.adminRenonceAdmin()`

Recommendation

We advise the client to use `_grantRole` instead of `grantRole`.

Alleviation

No alleviation.

GLOBAL-17 | Logic Issue Of Function `setWhitelist`

Category	Severity	Location	Status
Logical Issue	Informational	Global	Resolved

Description

The admin or whitelist manager can set whitelist but cannot remove it. We would like to confirm with the client if the current implementation aligns with the original project design. For example：

- `ERC20AssetHandler.setWhitelist()`
- `TrustlessAssetHandler.setWhitelist()`
- `RegistryHub.setWhitelist()`

Alleviation

The client heeded our advice and resolved this issue in commit : ea2e2eccebcda3da556015b5c273d9501496257b.

GLOBAL-18 | Discussion For Function `addPool()`

Category	Severity	Location	Status
Logical Issue	● Informational	Global	ⓘ Acknowledged

Description

According to the current logic of `StakingTemplate`, if the admin wants to add a staking pool and makes it work, he must invoke the function `adminDepositReward()` and `addPool()`. He must also be careful with the input `amount` when invoking `adminDepositReward()`. We want to know that if there's any further step for optimizing this. If we are missing anything, please let us know.

Alleviation

`[Nutbox]`: Actually, he will deploy a mintable token as his community token. In this way, he does not need to invoke adminDepositReward. And if he use simple `erc20`, he called `adminDepositRewad` to charge balance token to community. He also can call `adminWithdrawReward` to withdraw them out.

BNC-01 | Not Update `relayerCount`

Category	Severity	Location	Status
Logical Issue	Major	bridge/Bridge.sol (v1): 59	Resolved

Description

According to the `adminRemoveRelaye` function logic, the `relayerCount` should be accumulated in the `adminAddRelayer` function.

Recommendation

We advise the client to update the `relayerCount` in the `adminAddRelayer` function.

Alleviation

The client heeded our advice and resolved this issue in commit : 873aa232e1c5c23ae0453967f4109ba661fd5e7b.

BNC-02 | Logic Issue Of Function `cancelProposal`

Category	Severity	Location	Status
Logical Issue	Minor	bridge/Bridge.sol (v1): 146	Resolved

Description

According to the current logic, the function can change the status to cancelled if the proposal' status does not equal to cancel. In this way, non-existent proposals and executed proposals can also be cancelled. We would like to confirm with the client if the current implementation aligns with the original project design.

Alleviation

The client resolved this issue in commit : 873aa232e1c5c23ae0453967f4109ba661fd5e7b.

BNC-03 | Redundant Code

Category	Severity	Location	Status
Logical Issue	Informational	bridge/Bridge.sol (v1): 163	Resolved

Description

The condition `threshold <= 1` can be included in `proposal.ayeVotes >= threshold` .

Recommendation

We advise the client to remove the condition `threshold <= 1`.

Alleviation

The client heeded our advice and resolved this issue in commit :
873aa232e1c5c23ae0453967f4109ba661fd5e7b.

ENC-01 | Logic Issue Of Function `executeProposal`

Category	Severity	Location	Status
Logical Issue	Informational	bridge/Executor.sol (v1): 45	Acknowledged

Description

The following issues are to be confirmed:

1. There is no logical implementation when `extrinsicType != 0`.
2. Calling `unlockOrMintAsset` function in the `ERC721AssetHandler` contract when `assetType == 2`, but the `ERC721AssetHandler.sol` file has not the function.

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

Alleviation

No alleviation.

ERH-01 | Unrestricted Access on `fundERC20`

Category	Severity	Location	Status
Logical Issue	Major	asset/ERC20Helper.sol (v1): 22	Resolved

Description

The function `fundERC20` could be called by anyone. So once `owner` approved allowance to `address(this)`, anyone could transfer `owner`‘ tokens into the contract.

Recommendation

We advise the client to recheck the logic.

Alleviation

The client resolved this issue by removing the function `fundERC20` in commit : 873aa232e1c5c23ae0453967f4109ba661fd5e7b.

NDC-01 | Lack of Error Message

Category	Severity	Location	Status
Coding Style	Informational	NoDelegateCall.sol (v1): 19	Acknowledged

Description

`require` can be used to check for conditions and throw an exception if the condition is not met, in which case the descriptive error message provided by the developer will appear and help to tracking error and debugging.

Recommendation

We advise the client to add error messages.

Alleviation

No alleviation.

NUT-01 | Visibility Specifiers Missing

Category	Severity	Location	Status
Language Specific	Informational	NUTToken.sol: 17	Resolved

Description

The linked variable declaration does not have a visibility specifier explicitly set.

Recommendation

Inconsistencies in the default visibility the Solidity compilers impose can cause issues in the functionality of the codebase. We advise that visibility specifier for the linked variable is explicitly set.

Alleviation

The client heeded our advice and resolved this issue in commit : 0b3dab054fd74f623220960f5fe277ed1969bfe3.

NUT-02 | Address Type Could Be Indexed In Events

Category	Severity	Location	Status
Gas Optimization	Informational	NUTToken.sol: 19~20	Resolved

Description

It is recommended to add `indexed` keyword for parameters in events, which makes it easier for users to navigate event logs.

Recommendation

We advise the client to add keyword `indexed` in the declaration of events.

Alleviation

The client heeded our advice and resolved this issue in commit : 0b3dab054fd74f623220960f5fe277ed1969bfe3.

NUT-03 | Initial Token Distribution

Category	Severity	Location	Status
Centralization / Privilege	**Major**	NUTToken.sol: 37	Acknowledged

Description

`initialSupply` tokens were sent to the `owner` when deploying the contract. This could be a centralization risk as the deployer can distribute tokens without obtaining the consensus of the community.

Recommendation

We recommend the team to be transparent regarding the initial token distribution process.

Alleviation

No alleviation.

NUT-04 | Redundant Code

Category	Severity	Location	Status
Logical Issue	Informational	NUTToken.sol: 39	Resolved

Description

The `transferOpened` is a `bool` type and its initial value is `false`.

Recommendation

We advise the client to remove it.

Alleviation

The client heeded our advice and resolved this issue in commit : 0b3dab054fd74f623220960f5fe277ed1969bfe3.

NUT-05 | Centralization Risk

Category	Severity	Location	Status
Centralization / Privilege	**Major**	NUTToken.sol: 51, 56, 62, 68, 72	Acknowledged

Description

To bridge the gap in trust between the administrators need to express a sincere attitude regarding the considerations of the administrator team's anonymity.

The `MINTER_ROLE` has the responsibility to notify users about the following capabilities:

- mint uncapped tokens to any address through `mint()`

The `owner` has the responsibility to notify users about the following capabilities:

- set any address to whitelist through `setWhiteList()`
- remove any address from whitelist through `removeWhiteList()`
- enable transfer through `enableTransfer()`
- disable transfer through `disableTransfer()`

Recommendation

We advise the client to carefully manage the privileged account's private keys to avoid any potential risks of being hacked. In general, we strongly recommend centralized privileges or roles in the protocol to be improved via a decentralized mechanism or via smart-contract-based accounts with enhanced security practices, e.g. Multisignature wallets.

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

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key;
- Introduction of a DAO/governance/voting module to increase transparency and user involvement.

Alleviation

No alleviation.

STN-01 | Check Effect Interaction Pattern Violated

Category	Severity	Location	Status
Logical Issue	Minor	staking/StakingTemplate.sol (v1): 354, 402, 433, 215	Acknowledged

Description

The sequence of external call/transfer and storage manipulation must follow a check effect interaction pattern. For example:

- `internalWithdraw()`
- `withdrawPoolRewards()`
- `withdrawTotalRewards()`
- `tryWithdraw()`

Recommendation

We advise the client to adopt `nonReentrant` modifier from openzeppelin library to the functions to prevent any reentrancy issue or use the checks-effects-interactions pattern. (LINK)

Alleviation

No alleviation.

STN-02 | Lack of Pool Validity Checks

Category	Severity	Location	Status
Logical Issue	Informational	staking/StakingTemplate.sol (v1): 402, 466, 491	Acknowledged

Description

There is no sanity check to validate if a pool exists.

Recommendation

We advise the client to adopt following modifier `validatePoolByPid` to functions `withdrawPoolRewards()`, `getUserPendingRewards()`, `getUserStakedAmount()`.

```
modifier validatePoolByPid(uint256 _pid) {
    require (_pid < numberOfPools , "Pool does not exist") ;
    _;
}
```

Alleviation

No alleviation.

STN-03 | Add `require` In Function `tryWithdraw`

Category	Severity	Location	Status
Logical Issue	Minor	staking/StakingTemplate.sol (v1): 215	Acknowledged

Description

According to the current logic, the admin could call this method multiple times until all users get refunded. If all users are refunded, all logic would execute which is a waste of gas. So we could add a check to prevent it.

Recommendation

We advise the client to adopt as follows:

```
require(!openedPools[pid].canRemove, "pool have removed");
```

Alleviation

No alleviation.

STN-04 | Logic Issue Of Function `adminWithdrawReward`

Category	Severity	Location	Status
Logical Issue	Critical	staking/StakingTemplate.sol (v1): 151	Resolved

Description

The `_lockAsset` is used in the `adminDepositReward` function. So the `adminWithdrawReward` function should use `_unlockAsset` instead of `_lockAsset`.

Recommendation

We advise the client to adopt as follows:

```
    function adminWithdrawReward(uint256 amount) public onlyAdmin {
        _unlockAsset(keccak256(abi.encodePacked(address(this), rewardAsset,
bytes("admin"))),
        rewardAsset, msg.sender, amount);
    }
```

Alleviation

The client heeded our advice and resolved this issue in commit : 873aa232e1c5c23ae0453967f4109ba661fd5e7b.

STN-05 | Add `require` In Function `removePool`

Category	Severity	Location	Status
Logical Issue	Minor	staking/StakingTemplate.sol (v1): 199	Acknowledged

Description

According to the current logic, the admin can remove a pool. If the pool has been removed, which is a waste of gas. So we could add a check to prevent it.

Recommendation

We advise the client to adopt as follows:

```
require(!openedPools[pid].hasRemoved, "pool have removed");
```

Alleviation

No alleviation.

Appendix

Finding Categories

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.

Gas Optimization

Gas Optimization findings do not affect the functionality of the code but generate different, more optimal EVM opcodes resulting in a reduction on the total gas cost of a transaction.

Logical Issue

Logical Issue findings detail a fault in the logic of the linked code, such as an incorrect notion on how block.timestamp 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.

Language Specific

Language Specific findings are issues that would only arise within Solidity, i.e. incorrect usage of private or delete.

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

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.

About

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.