BOT Chain - Chain

Preliminary Comments

CertiK Assessed on Feb 23rd, 2026



CertiK Assessed on Feb 23rd, 2026

BOT Chain - Chain

These preliminary comments were prepared by CertiK.

Executive Summary

TYPES	ECOSYSTEM	METHODS
DEX, Staking	Binance Smart Chain (BSC)	Manual Review, Static Analysis

LANGUAGE	TIMELINE
Solidity	Preliminary comments published on 02/09/2026

Vulnerability Summary



8	3	0	5	0	0
Total Findings	Resolved	Partially Resolved	Acknowledged	Declined	Pending

3	Centralization	3 Acknowledged	Centralization findings highlight privileged roles & functions and their capabilities, or instances where the project takes custody of users' assets.
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.
0	Major		Major risks may include logical errors that, under specific circumstances, could result in fund losses or loss of project control.
0	Medium		Medium risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform.
2	Minor	2 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.
3	Informational	3 Resolved	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.
0	Discussion		The impact of the issue is yet to be determined, hence requires further clarifications from the project team.

TABLE OF CONTENTS | BOT CHAIN - CHAIN

Summary

Executive Summary

Vulnerability Summary

Codebase

Audit Scope

Approach & Methods

System Overview

BOT Staking & Chain Governance (BSC Fork)

Review Notes

Out-of-Scope Dependencies

Testing & Documentation

Code Completeness

Findings

BCC-01 : Hidden Privileged Role In `SystemV2`

BCC-02 : Centralized Control Of Contract Upgrade

BCC-03 : Centralizatio Risks

BCC-04 : Missing Zero Address Validation

BCC-05 : Potential Cross-Chain Replay Attack With Merkle Proofing Mechanism

BCC-06 : Contracts With Todos

BCC-07 : Inconsistency Between Comment And Code

BCC-08 : Commented Out Code

Appendix

Disclaimer

CODEBASE | BOT CHAIN - CHAIN

Repository

Private Repo

AUDIT SCOPE | BOT CHAIN - CHAIN

CertiKProject/certik-audit-projects
BSCGovernor.sol
BSCValidatorSet.sol
SlashIndicator.sol
StakeHub.sol
System.sol

APPROACH & METHODS | BOT CHAIN - CHAIN

This audit was conducted for BOT Chain to evaluate the security and correctness of the smart contracts associated with the BOT Chain - Chain project. The assessment included a comprehensive review of the in-scope smart contracts. The audit was performed using a combination of Manual Review and Static Analysis.

The review process emphasized the following areas:

- Architecture review and threat modeling to understand systemic risks and identify design-level flaws.
- Identification of vulnerabilities through both common and edge-case attack vectors.
- Manual verification of contract logic to ensure alignment with intended design and business requirements.
- Dynamic testing to validate runtime behavior and assess execution risks.
- Assessment of code quality and maintainability, including adherence to current best practices and industry standards.

The audit resulted in findings categorized across multiple severity levels, from informational to critical. To enhance the project's security and long-term robustness, we recommend addressing the identified issues and considering the following general improvements:

- Improve code readability and maintainability by adopting a clean architectural pattern and modular design.
- Strengthen testing coverage, including unit and integration tests for key functionalities and edge cases.
- Maintain meaningful inline comments and documentations.
- Implement clear and transparent documentation for privileged roles and sensitive protocol operations.
- Regularly review and simulate contract behavior against newly emerging attack vectors.

SYSTEM OVERVIEW | BOT CHAIN - CHAIN

The BOT Chain audit covers a chain governance and staking suite based on BSC chain.

BOT Staking & Chain Governance (BSC Fork)

The chain governance suite includes `System`, `StakeHub`, `BSCValidatorSet`, `SlashIndicator`, and `BSCGovernor`. **Note that for this part, only the differences between the BOT chain contracts and the `bsc-genesis-contract` have been audited**.

- `System` defines fixed system addresses and core access-control modifiers for consensus, cross-chain, slashing, and governance actors.
- `StakeHub` manages validator creation, delegation, redelegation, unbonding, and slashing-related parameters.
- `BSCValidatorSet` maintains the active validator set, handles fee distribution and maintenance state, and integrates with system reward logic.
- `SlashIndicator` tracks validator downtime and malicious behavior, enforces slash thresholds, and signals felony/misdemeanor events.
- `BSCGovernor` is an OpenZeppelin Governor-based governance module with a timelock, proposal threshold/quorum rules, and a whitelist of targets.

REVIEW NOTES | BOT CHAIN - CHAIN

Out-of-Scope Dependencies

The codebase contains multiple components and dependencies that were not fully covered by this assessment. In particular:

- Parts of the repository are **forked or derived from upstream projects** (e.g., Uniswap V3 core/periphery). Components that were not explicitly in scope and/or appear unchanged from upstream were treated as external dependencies and assumed to behave as specified by their original maintainers.
- **Third-party libraries and build tooling** were not audited comprehensively. Any security properties of the development toolchain are outside the scope of this report.
- The security of **external contracts and external systems** interacted with by the reviewed contracts (e.g., ERC-20 tokens, WETH, bridge operators/relayers, off-chain signing infrastructure, governance processes, and key management) is out of scope. These elements were treated as operational assumptions unless the contract enforces the relevant security guarantees on-chain.

Our review focused on the on-chain behavior of the in-scope Solidity contracts within the defined scope.

Testing & Documentation

The repository includes automated tests and developer tooling, particularly around the Uniswap-derived components. However, overall **integration and edge-case coverage appears limited** across the full system, especially when considering cross-component behavior (e.g., bridge flows, privileged operations, and configuration correctness).

We recommend expanding test coverage to include stress scenarios and failure modes (e.g., misconfiguration, unexpected inputs, privilege/role transitions, and revert-path behavior). On the documentation front, inline comments exist, but higher-level documentation is limited. Adding concise system-level documentation (architecture and operational assumptions) would improve maintainability and reduce misconfiguration risk.

Code Completeness

We noticed several `TODO` / `FIXME` comments remaining in the codebase. These markers may indicate incomplete hardening work, deferred changes, or leftover development notes. We recommend triaging these items by resolving or removing outdated TODO/FIXME notes and re-reviewing any security-relevant logic added or modified as a result.

FINDINGS | BOT CHAIN - CHAIN



8	0	3	0	0	2	3	0
Total Findings	Critical	Centralization	Major	Medium	Minor	Informational	Discussion

This report has been prepared for BOT Chain to identify potential vulnerabilities and security issues within the reviewed codebase. During the course of the audit, a total of 8 issues were identified. Leveraging a combination of Manual Review & Static Analysis the following findings were uncovered:

ID	Title	Category	Severity	Status
BCC-01	**Hidden Privileged Role In `SystemV2`**	**Centralization**	**Centralization**	**Acknowledged**
BCC-02	**Centralized Control Of Contract Upgrade**	**Centralization**	**Centralization**	**Acknowledged**
BCC-03	**Centralizatio Risks**	**Centralization**	**Centralization**	**Acknowledged**
BCC-04	Missing Zero Address Validation	Volatile Code	Minor	Acknowledged
BCC-05	Potential Cross-Chain Replay Attack With Merkle Proofing Mechanism	Design Issue	Minor	Acknowledged
BCC-06	Contracts With Todos	Coding Issue	Informational	Resolved
BCC-07	Inconsistency Between Comment And Code	Inconsistency	Informational	Resolved
BCC-08	Commented Out Code	Coding Style	Informational	Resolved

BCC-01 | Hidden Privileged Role In `SystemV2`

Category	Severity	Location	Status
Centralization	Centralization	SystemV2.sol (base): 51, 56, 61, 66, 71, 76, 81	Acknowledged

Description

`SystemV2` restricts certain functions to predefined system contract addresses (e.g., GovHub, Slash, StakeHub) by checking `msg.sender`. However, these system addresses are declared as `internal constant` and are not exposed via public getter functions, which may reduce transparency and cause confusion for integrators/users relying on the ABI.

Recommendation

Change the visibility of the private role or add the `getter` function to clarify the transparency or clarify the intended behavior.

Alleviation

[BOT Chain, 02/23/2026]: The team acknowledged this issue.

[CertiK, 02/23/2026]: 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.

BCC-02 | Centralized Control Of Contract Upgrade

Category	Severity	Location	Status
Centralization	Centralization	BSCGovernor.sol (base): 17; BSCTimelock.sol (base): 9; GovToken.sol (base): 12; StakeCredit.sol (base): 12; TokenRecoverPortal.sol (base): 21	Acknowledged

Description

In the contracts `BSCGovernor`, `BSCTimelock`, `GovToken`, `StakeCredit`, and `TokenRecoverPortal`, the privileged account `proxy admin` has the authority to update the implementation contract behind the upgradeable proxy.

Any compromise of the `proxy admin` may allow a hacker to upgrade the proxy to a malicious implementation, thereby changing contract logic and potentially enabling arbitrary minting, fund draining, censorship of governance actions, or other malicious behaviors depending on the upgraded implementation.

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

[BOT Chain, 02/23/2026]: The team acknowledged this issue.

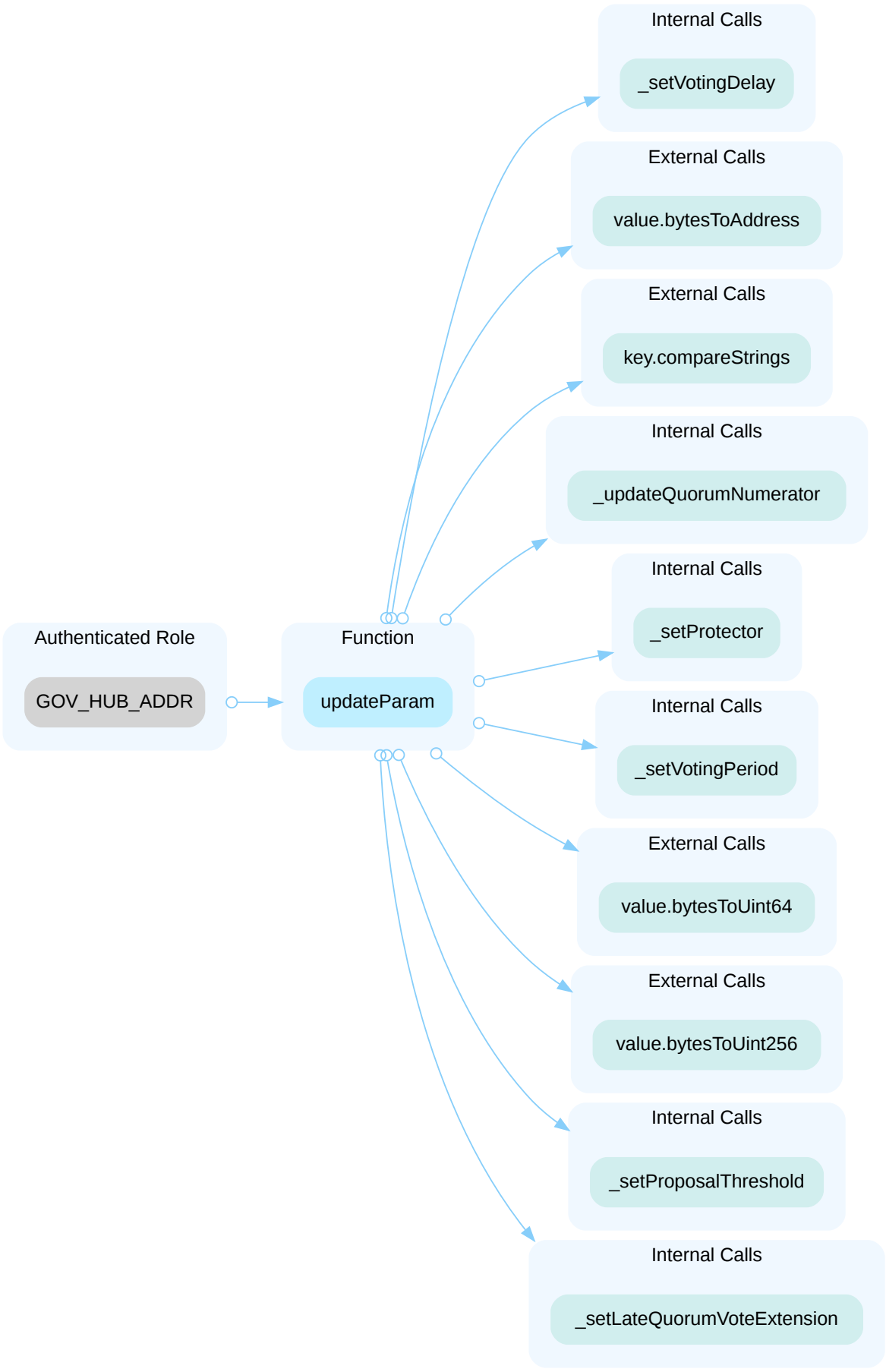
[CertiK, 02/23/2026]: It is suggested to implement the aforementioned methods to avoid centralized failure. Also, CertiK strongly encourages the project team to periodically revisit the private key security management of all addresses related to centralized roles.

BCC-03 | Centralizatio Risks

Category	Severity	Location	Status
Centralization	Centralization	BSCGovernor.sol (base): 171; BSCTimelock.sol (base): 31; BSCValidatorSet.sol (base): 168, 175, 179, 521, 602; GovHub.sol (base): 21, 29, 34, 38; GovToken.sol (base): 50, 59, 71; SlashIndicator.sol (base): 90, 97, 101, 137, 193, 315, 357; StakeCredit.sol (base): 73, 85, 103, 114, 136, 151, 188, 211; SystemReward.sol (base): 42, 63; TokenHub.sol (base): 106, 113, 130, 141, 154, 190, 201, 238, 300, 304; TokenRecoverPortal.sol (base): 243	Acknowledged

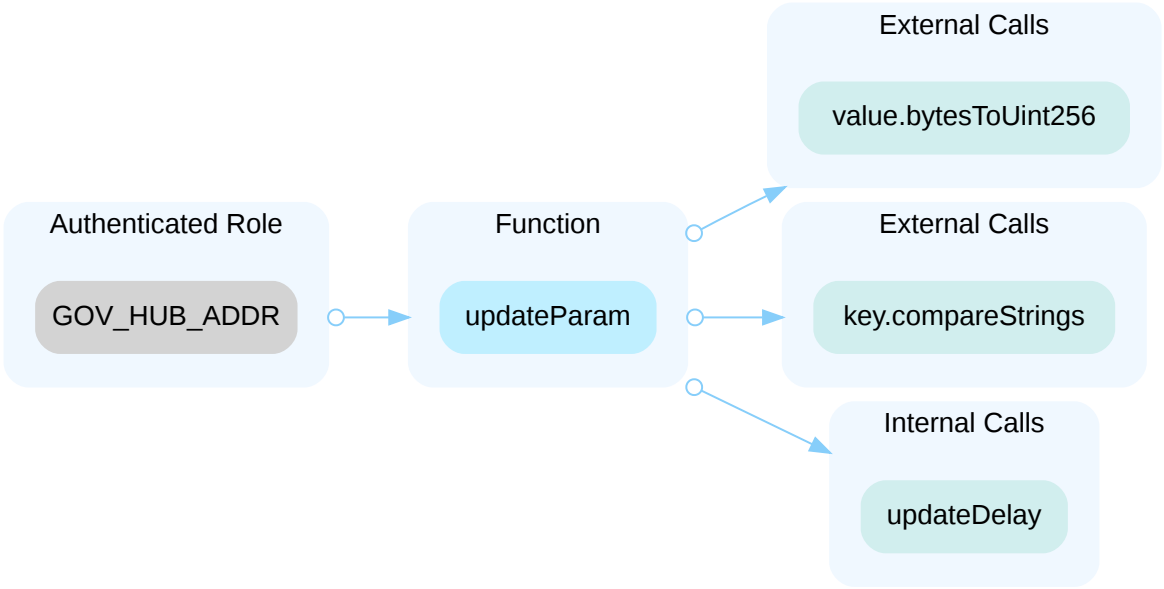
Description

In the contract `BSCGovernor` , the role `GOV_HUB_ADDR` has authority over the functions shown in the diagram below. Any compromise to the `GOV_HUB_ADDR` account may allow the hacker to take advantage of this authority and change critical governance parameters (e.g., voting delay/period, proposal threshold, quorum) and the governor protector.

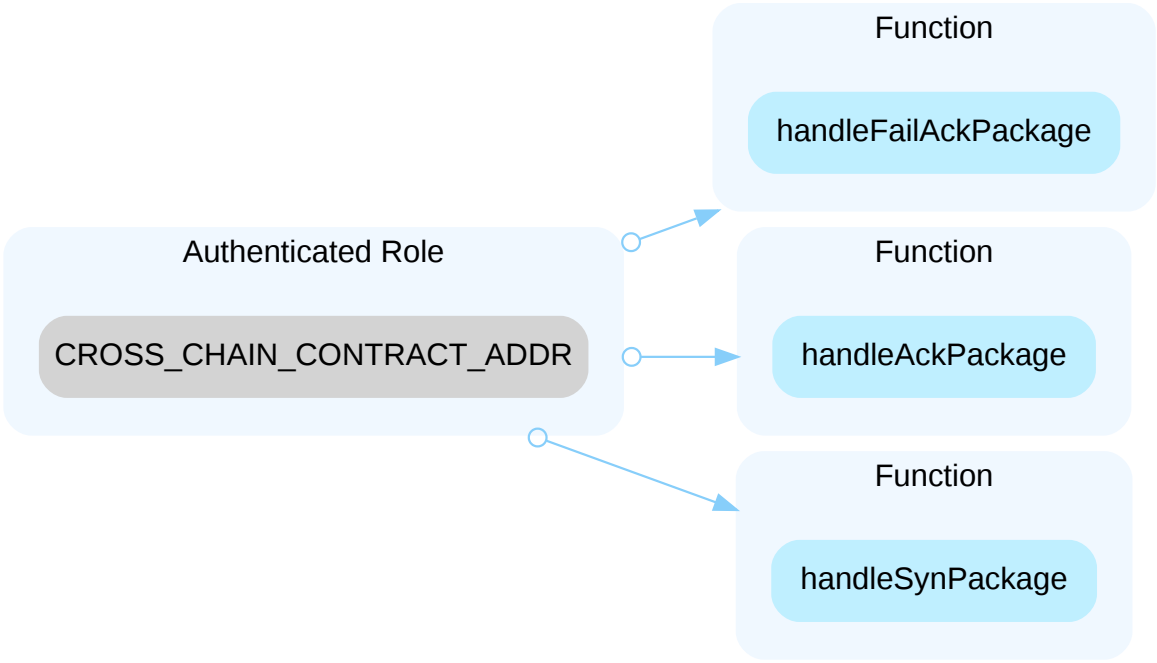


CERTIK PRELIMINARY COMMENTS

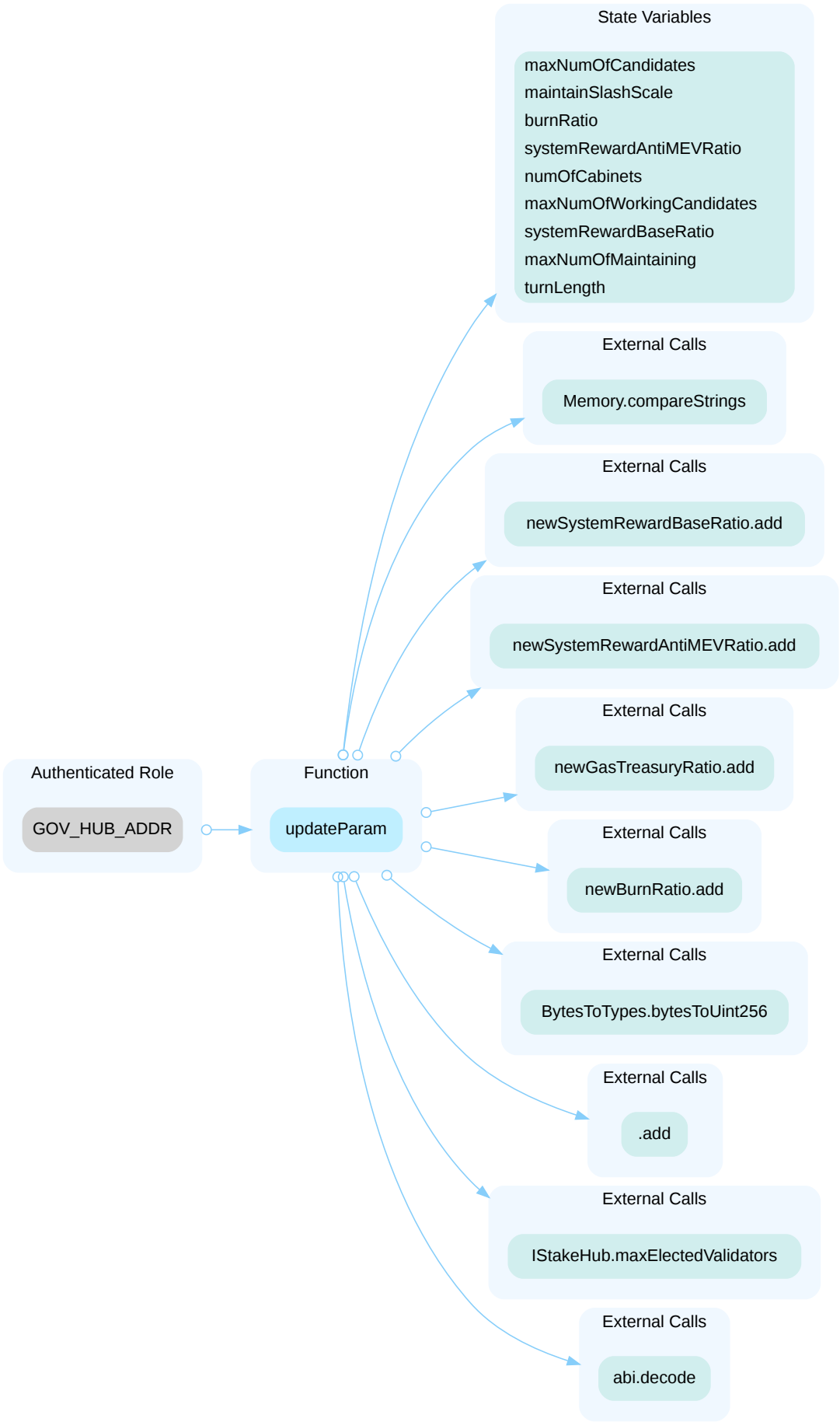
In the contract `BSCTimelock`, the role `GOV_HUB_ADDR` has authority over the functions shown in the diagram below. Any compromise to the `GOV_HUB_ADDR` account may allow the hacker to take advantage of this authority and change the timelock minimum delay (e.g., reduce the delay).



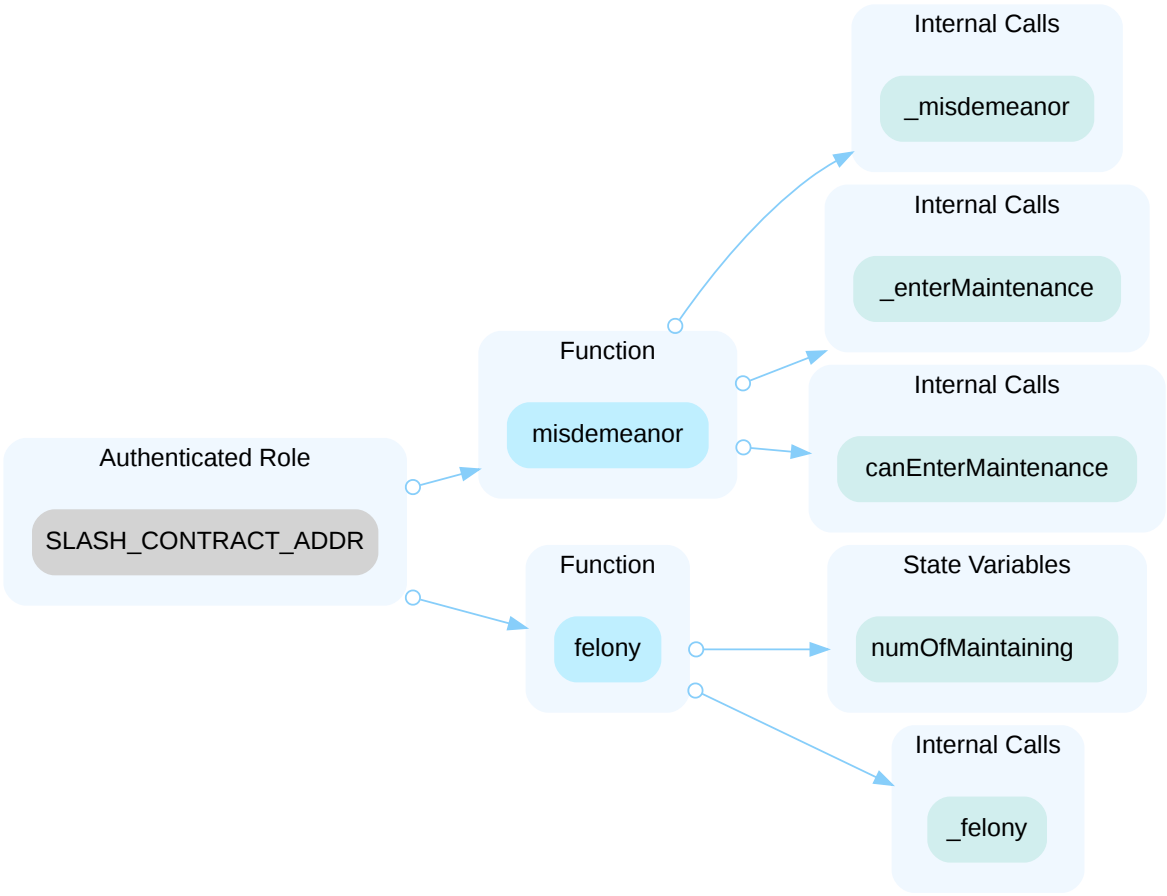
In the contract `BSCValidatorSet`, the role `CROSS_CHAIN_CONTRACT_ADDR` has authority over the functions shown in the diagram below. Any compromise to the `CROSS_CHAIN_CONTRACT_ADDR` account may allow the hacker to take advantage of this authority and call cross-chain handler entrypoints (which are deprecated and revert in this version).



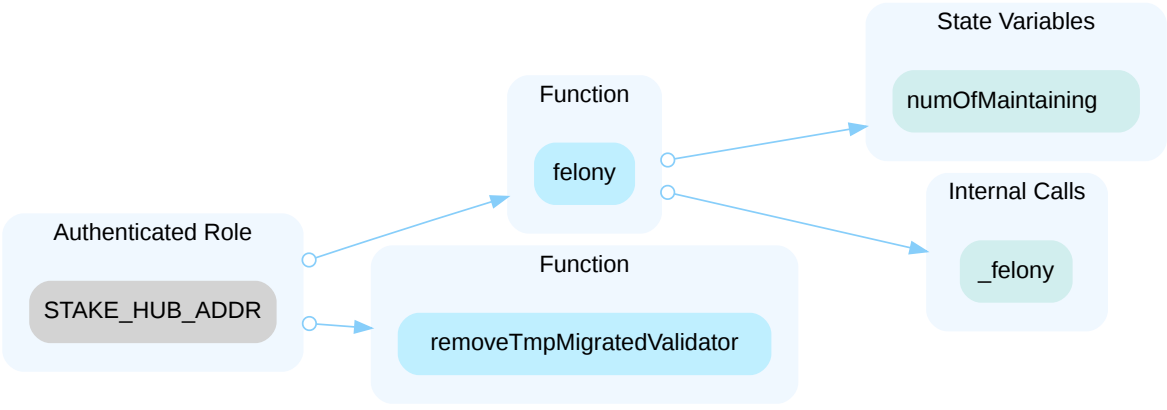
In the contract `BSCValidatorSet`, the role `GOV_HUB_ADDR` has authority over the functions shown in the diagram below. Any compromise to the `GOV_HUB_ADDR` account may allow the hacker to take advantage of this authority and change critical consensus and fee-distribution parameters (e.g., burn ratio, cabinet size, gas treasury settings).


State Variables
maxNumOfCandidates
maintainSlashScale
burnRatio
systemRewardAntiMEVRatio
numOfCabinets
maxNumOfWorkingCandidates
systemRewardBaseRatio
maxNumOfMaintaining
turnLength
External Calls
Memory.compareStrings
External Calls
newSystemRewardBaseRatio.add
External Calls
newSystemRewardAntiMEVRatio.add
External Calls
newGasTreasuryRatio.add
External Calls
newBurnRatio.add
External Calls
BytesToTypes.bytesToUint256
External Calls
.add
External Calls
IStakeHub.maxElectedValidators
External Calls
abi.decode
Authenticated Role
GOV_HUB_ADDR
Function
updateParam

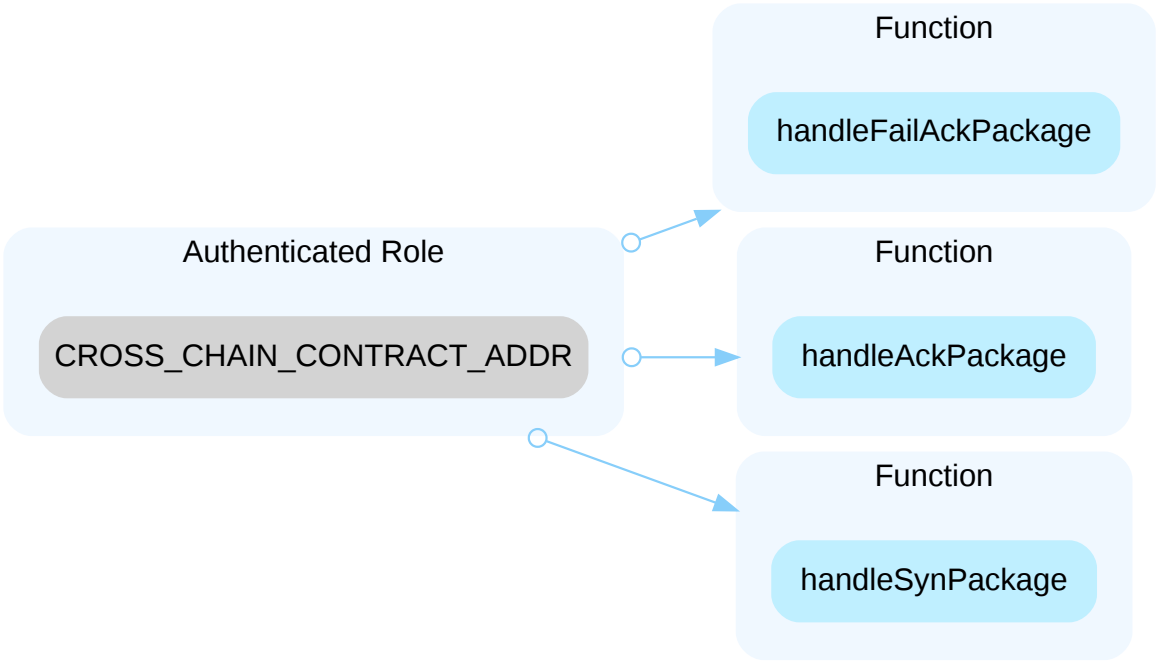
In the contract `BSCValidatorSet`, the role `SLASH_CONTRACT_ADDR` has authority over the functions shown in the diagram below. Any compromise to the `SLASH_CONTRACT_ADDR` account may allow the hacker to take advantage of this authority and arbitrarily misdemeanor/felony slash (jail/remove) validators.



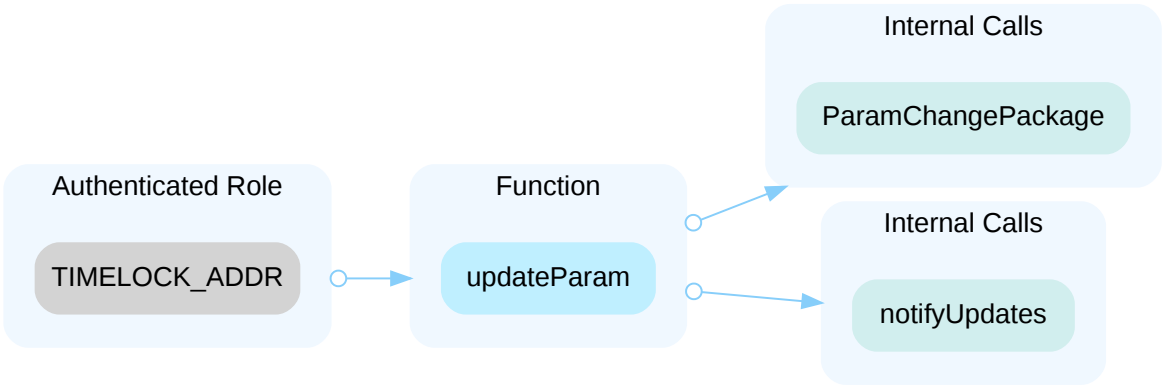
In the contract `BSCValidatorSet`, the role `STAKE_HUB_ADDR` has authority over the functions shown in the diagram below. Any compromise to the `STAKE_HUB_ADDR` account may allow the hacker to take advantage of this authority and trigger felony slashing/removal of validators.



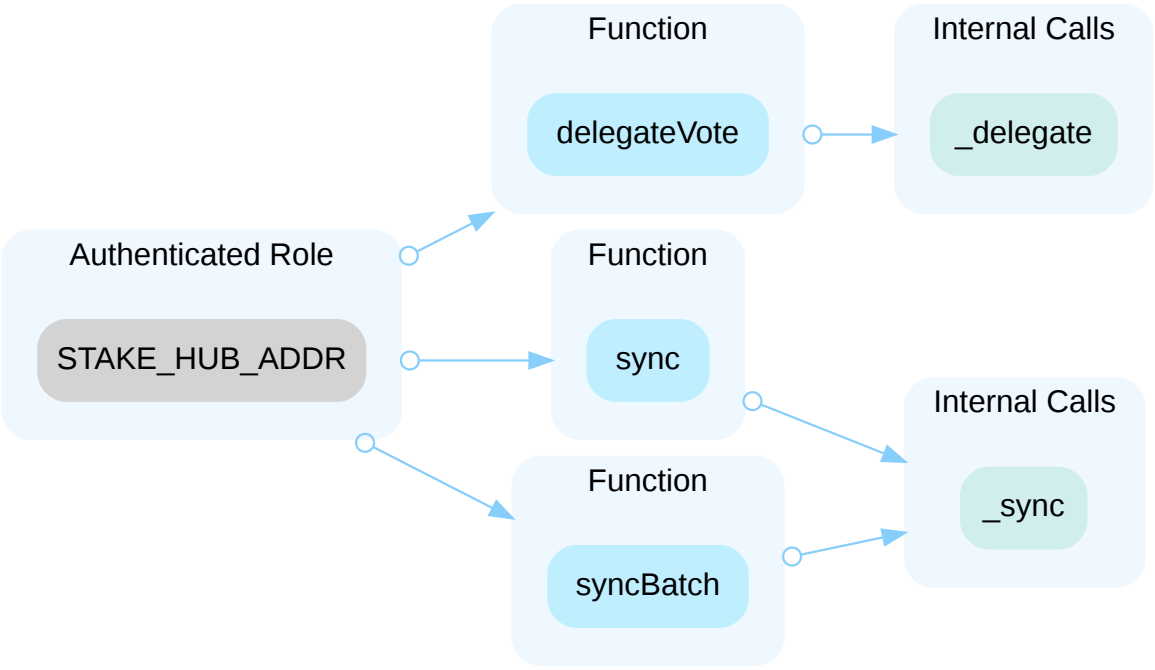
In the contract `GovHub`, the role `CROSS_CHAIN_CONTRACT_ADDR` has authority over the functions shown in the diagram below. Any compromise to the `CROSS_CHAIN_CONTRACT_ADDR` account may allow the hacker to take advantage of this authority and call cross-chain handler entrypoints (which are deprecated and revert in this version).



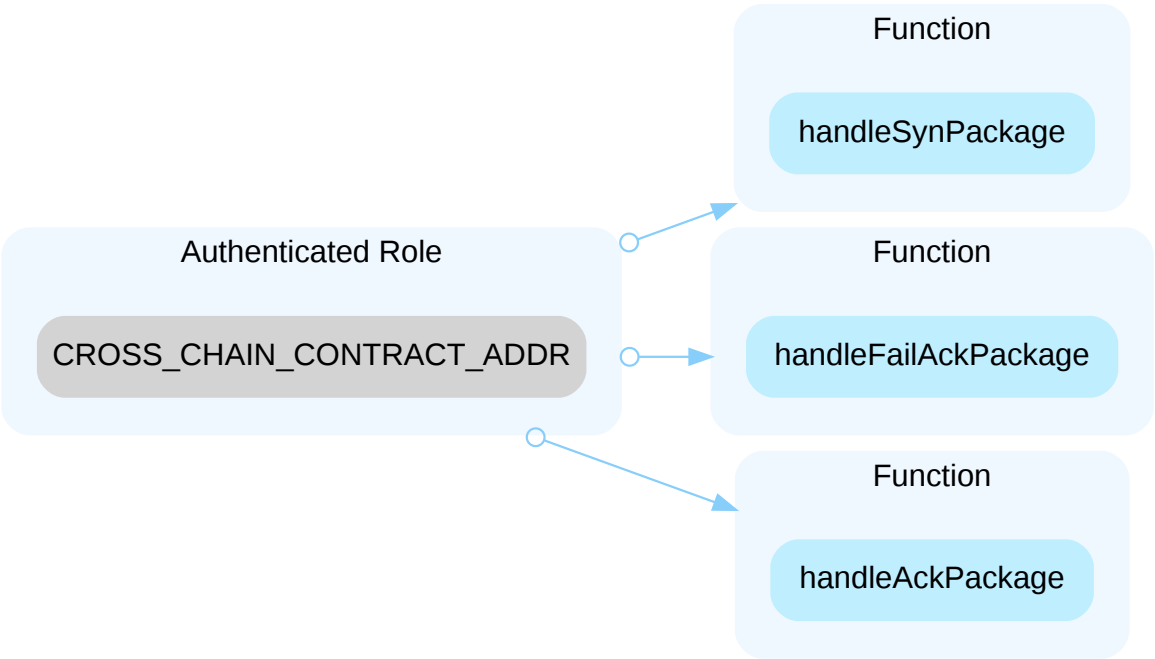
In the contract `GovHub`, the role `TIMELOCK_ADDR` has authority over the functions shown in the diagram below. Any compromise to the `TIMELOCK_ADDR` account may allow the hacker to take advantage of this authority and push arbitrary `updateParam(...)` calls to system contracts via `GovHub.updateParam(...)`.



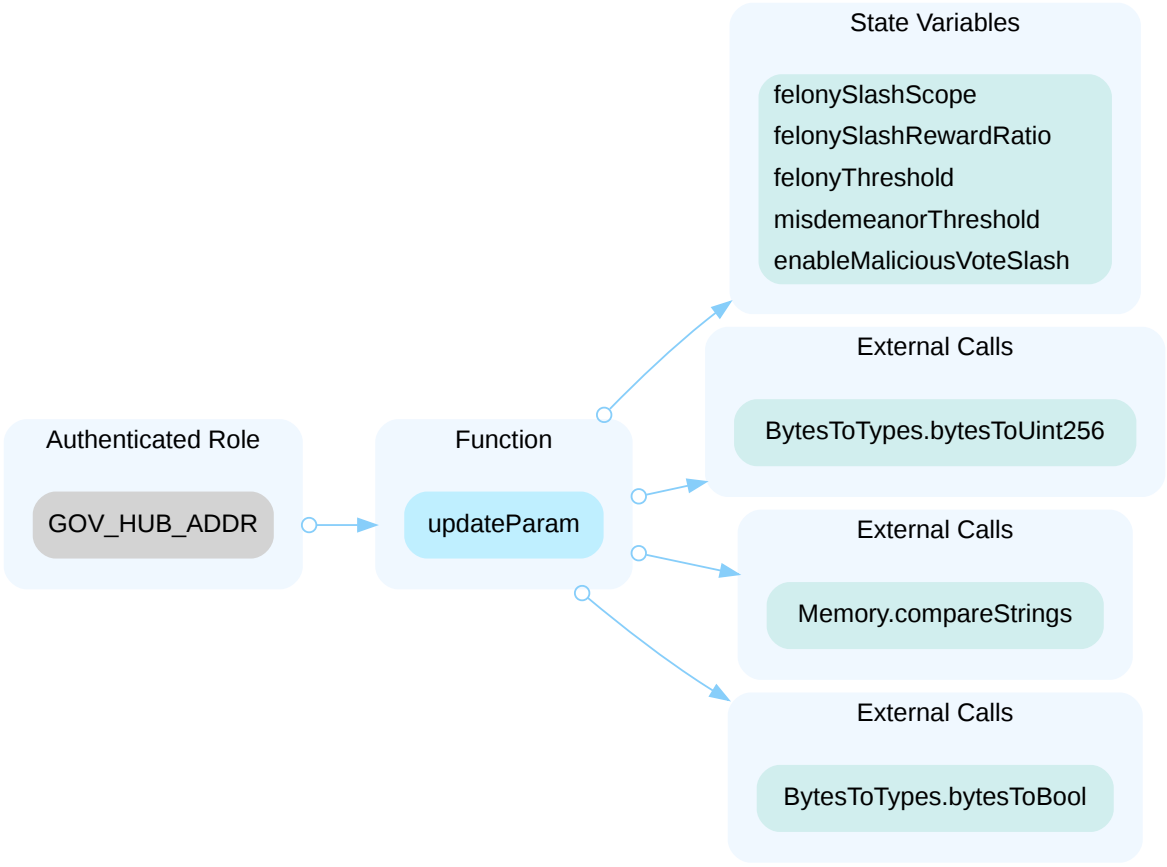
In the contract `GovToken`, the role `STAKE_HUB_ADDR` has authority over the functions shown in the diagram below. Any compromise to the `STAKE_HUB_ADDR` account may allow the hacker to take advantage of this authority and mint/burn `govBNB` (via `sync`) and delegate votes.



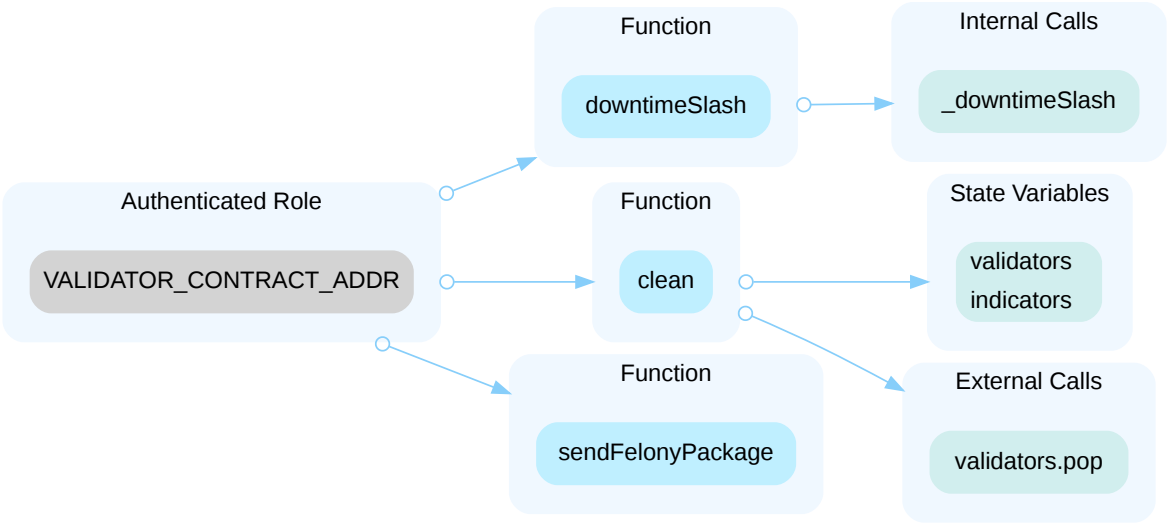
In the contract `SlashIndicator`, the role `CROSS_CHAIN_CONTRACT_ADDR` has authority over the functions shown in the diagram below. Any compromise to the `CROSS_CHAIN_CONTRACT_ADDR` account may allow the hacker to take advantage of this authority and call cross-chain handler entrypoints (which are deprecated and revert in this version).



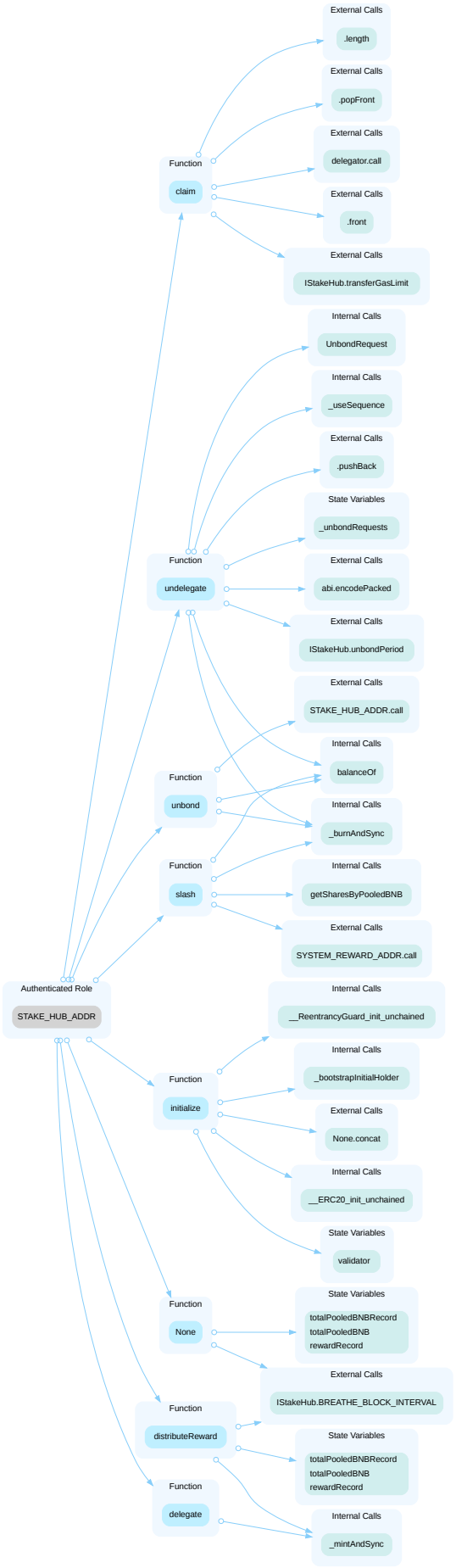
In the contract `SlashIndicator`, the role `GOV_HUB_ADDR` has authority over the functions shown in the diagram below. Any compromise to the `GOV_HUB_ADDR` account may allow the hacker to take advantage of this authority and change slashing thresholds/settings (including enabling malicious-vote slashing).



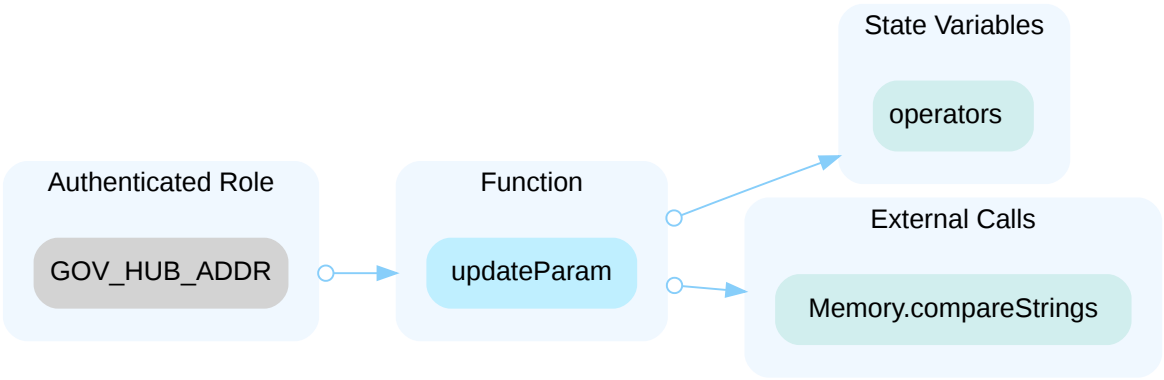
In the contract `SlashIndicator`, the role `VALIDATOR_CONTRACT_ADDR` has authority over the functions shown in the diagram below. Any compromise to the `VALIDATOR_CONTRACT_ADDR` account may allow the hacker to take advantage of this authority and trigger downtime slashing actions against validators.



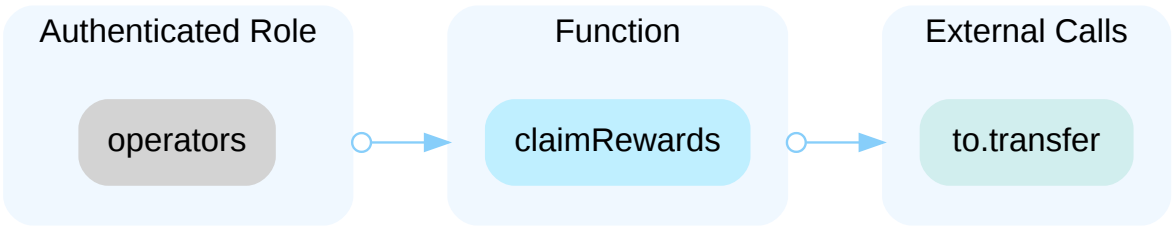
In the contract `StakeCredit`, the role `STAKE_HUB_ADDR` has authority over the functions shown in the diagram below. Any compromise to the `STAKE_HUB_ADDR` account may allow the hacker to take advantage of this authority and burn users' shares and withdraw underlying BNB to `StakeHub` (e.g., via `unbond`).


Authenticated Role
STAKE_HUB_ADDR
Function
claim
External Calls
.length
External Calls
.popFront
External Calls
delegator.call
External Calls
.front
External Calls
IStakeHub.transferGasLimit
Function
undelegate
Internal Calls
UnbondRequest
Internal Calls
_useSequence
External Calls
.pushBack
State Variables
_unbondRequests
External Calls
abi.encodePacked
External Calls
IStakeHub.unbondPeriod
Function
unbond
External Calls
STAKE_HUB_ADDR.call
Internal Calls
balanceOf
Internal Calls
_burnAndSync
Function
slash
Internal Calls
getSharesByPooledBNB
External Calls
SYSTEM_REWARD_ADDR.call
Function
initialize
Internal Calls
__ReentrancyGuard_init_unchained
Internal Calls
_bootstrapInitialHolder
External Calls
None.concat
Internal Calls
__ERC20_init_unchained
State Variables
validator
Function
None
State Variables
totalPooledBNBRecord
totalPooledBNB
rewardRecord
Function
distributeReward
External Calls
IStakeHub.BREATHE_BLOCK_INTERVAL
State Variables
totalPooledBNBRecord
totalPooledBNB
rewardRecord
Function
delegate
Internal Calls
_mintAndSync

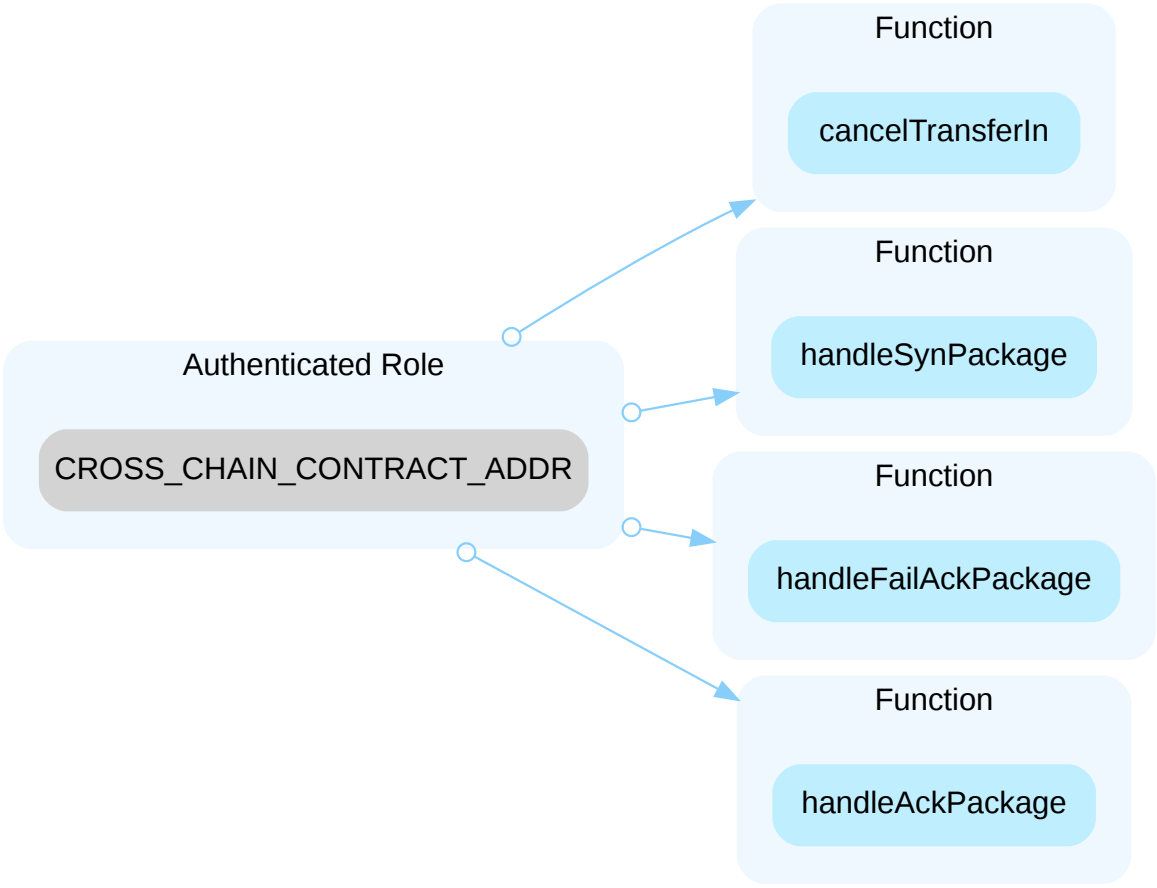
In the contract `SystemReward`, the role `GOV_HUB_ADDR` has authority over the functions shown in the diagram below. Any compromise to the `GOV_HUB_ADDR` account may allow the hacker to take advantage of this authority and add/remove `operators` that can withdraw rewards.



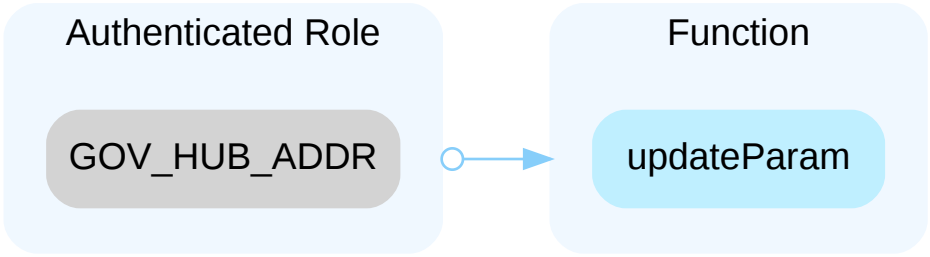
In the contract `SystemReward`, the role `operators` has authority over the functions shown in the diagram below. Any compromise to the `operators` account may allow the hacker to take advantage of this authority and claim/withdraw SystemReward funds to arbitrary recipients (up to the per-call cap).



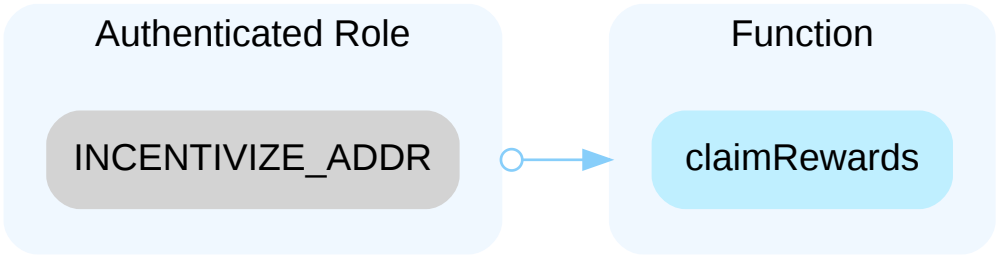
In the contract `TokenHub`, the role `CROSS_CHAIN_CONTRACT_ADDR` has authority over the functions shown in the diagram below. Any compromise to the `CROSS_CHAIN_CONTRACT_ADDR` account may allow the hacker to take advantage of this authority and exercise those privileges.



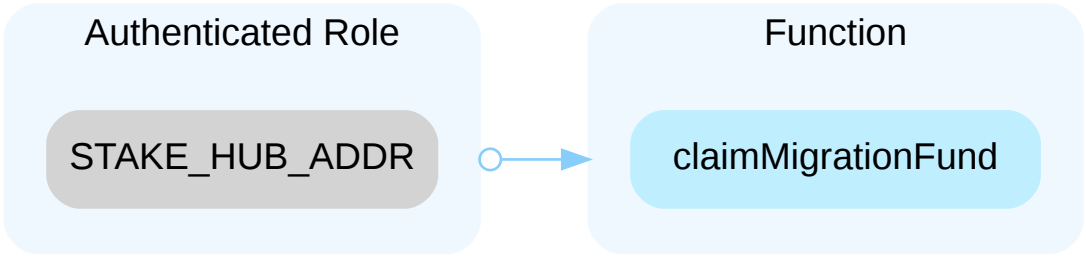
In the contract `TokenHub`, the role `GOV_HUB_ADDR` has authority over the functions shown in the diagram below. Any compromise to the `GOV_HUB_ADDR` account may allow the hacker to take advantage of this authority and call the governance parameter update entrypoint (which is deprecated and reverts in this version).



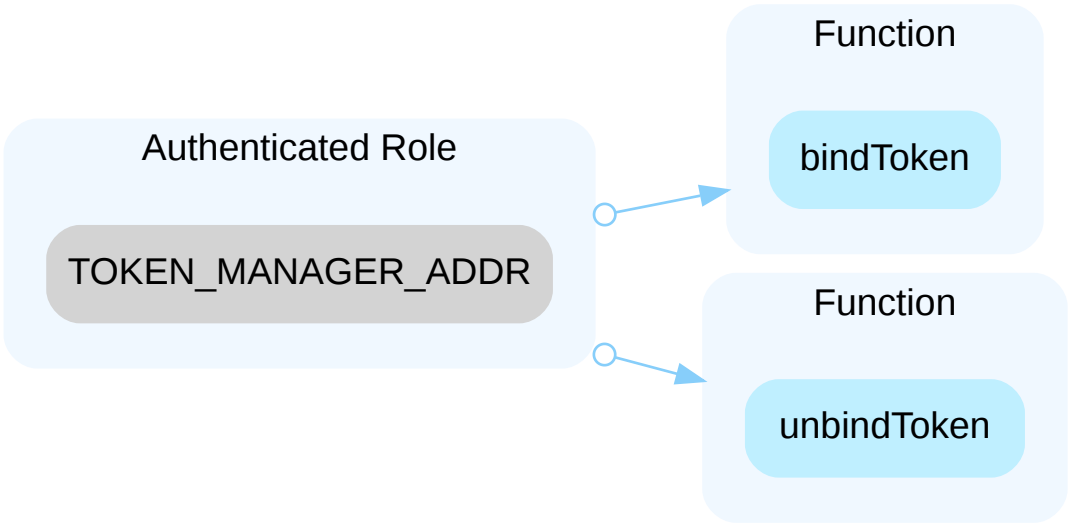
In the contract `TokenHub`, the role `INCENTIVIZE_ADDR` has authority over the functions shown in the diagram below. Any compromise to the `INCENTIVIZE_ADDR` account may allow the hacker to take advantage of this authority and call the relayer-reward entrypoint (which is deprecated and reverts in this version).



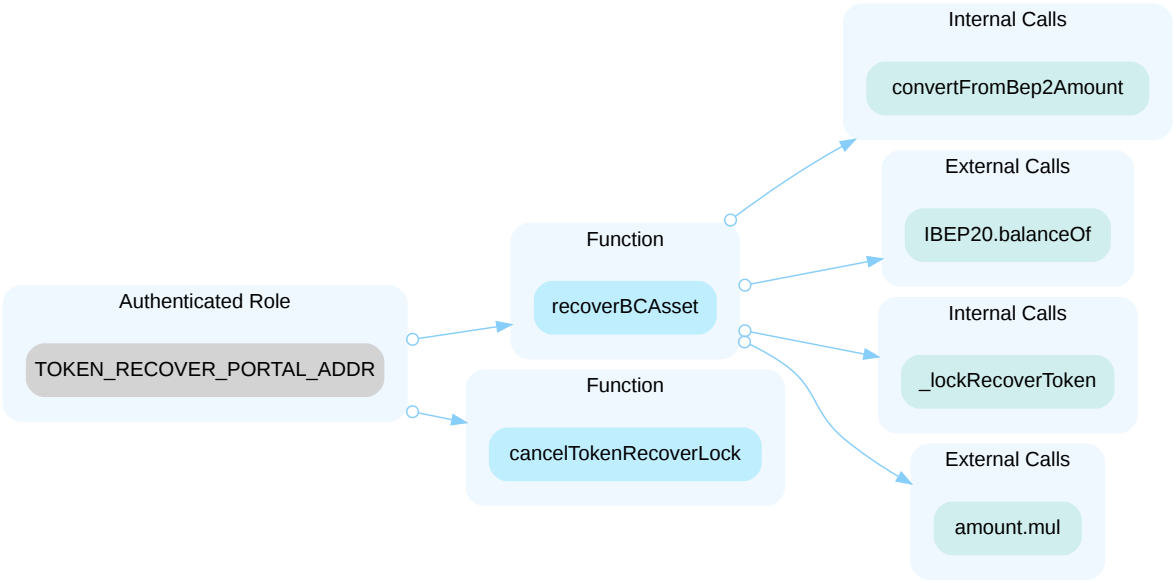
In the contract `TokenHub`, the role `STAKE_HUB_ADDR` has authority over the functions shown in the diagram below. Any compromise to the `STAKE_HUB_ADDR` account may allow the hacker to take advantage of this authority and call the migration-fund claim entrypoint (which is deprecated and reverts in this version).



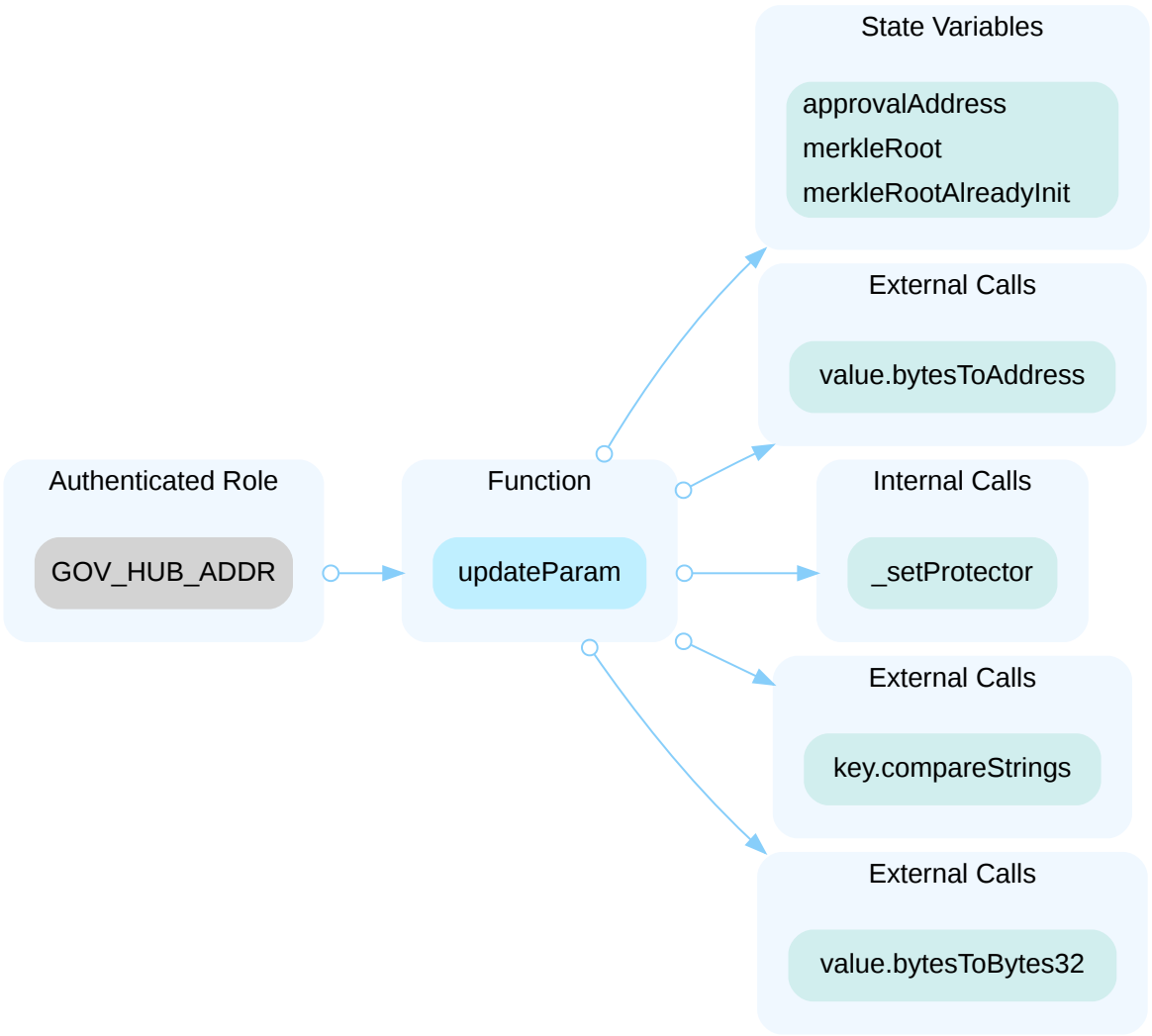
In the contract `TokenHub`, the role `TOKEN_MANAGER_ADDR` has authority over the functions shown in the diagram below. Any compromise to the `TOKEN_MANAGER_ADDR` account may allow the hacker to take advantage of this authority and call token bind/unbind entrypoints (which are deprecated and revert in this version).



In the contract `TokenHub`, the role `TOKEN_RECOVER_PORTAL_ADDR` has authority over the functions shown in the diagram below. Any compromise to the `TOKEN_RECOVER_PORTAL_ADDR` account may allow the hacker to take advantage of this authority and lock TokenHub assets for arbitrary recipients and cancel existing recovery locks.



In the contract `TokenRecoverPortal`, the role `GOV_HUB_ADDR` has authority over the functions shown in the diagram below. Any compromise to the `GOV_HUB_ADDR` account may allow the hacker to take advantage of this authority and set `approvalAddress` / initialize `merkleRoot` and change the protector.



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

[BOT Chain, 02/23/2026]: The team acknowledged this issue.

[CertiK, 02/23/2026]: 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.

BCC-04 | Missing Zero Address Validation

Category	Severity	Location	Status
Volatile Code	Minor	StakeCredit.sol (base): 91	Acknowledged

Description

The cited address input is missing a check that it is not `address(0)` .

Recommendation

We recommend adding a check the passed-in address is not `address(0)` to prevent unexpected errors.

Alleviation

[BOT Chain, 02/17/2026]: The team acknowledged the issue and decided not to implement the recommended change in the current engagement. Moreover, the initialization flow is invoked through `IStakeCredit(creditProxy).initialize{value: msg.value}(operatorAddress, moniker)` , and the team currently does not consider the zero address risk to be applicable in this context, pending further internal confirmation.

BCC-05 Potential Cross-Chain Replay Attack With Merkle Proofing Mechanism

Category	Severity	Location	Status
Design Issue	Minor	TokenRecoverPortal.sol (base): 156	Acknowledged

Description

Some contracts are intended to be deployed across chains. If the contract enables the Merkle proofing mechanism, there should be protection from using the same `root` of the Merkle tree in different chains.

If there is no such protection, since signed messages are not properly verified with the current chain ID, thus the contract allows attackers to perform replay attacks across chains.

Recommendation

To mitigate the risk of cross-chain replay attacks, consider the following approaches:

- **Use Chain-Specific Merkle Roots**: Generate unique Merkle roots for vesting contracts deployed on different chains.
- **Include Chain Identifiers**: Incorporate chain-specific data (such as chain id) into the user allocation data or the Merkle proof to ensure that claims are only valid on the intended chain.
- **Contract-Level Chain Checks**: Implement checks within the contract to verify the chain on which it is operating, if supported by the underlying blockchain platform.

Alleviation

[BOT Chain, 02/17/2026]: The team acknowledged the issue and decided not to implement the recommended change in the current engagement. Moreover, the referenced contract is not planned to be used in the current deployment scope, and therefore no immediate remediation is scheduled.

BCC-06 | Contracts With Todos

Category	Severity	Location	Status
Coding Issue	Informational	BSCGovernor.sol (base): 37; BSCValidatorSet.sol (base): 620	Resolved

Description

The codebase contains TODO comments inside production smart contracts. While comments do not affect on-chain execution, they may indicate incomplete development work or leftover debug references that were not fully cleaned up before release.

In particular, some TODO comments reference alternative (and potentially unsafe) implementations (e.g., a commented function signature that would remove critical access control). If such changes are later introduced post-audit or accidentally enabled during refactoring, they may bypass intended permission checks and create high-impact vulnerabilities that were not evaluated in the audited version.

Recommendation

We recommend removing or updating TODO comments in production contracts, especially those that reference insecure or debug-only logic. If the team plans to implement any functionality hinted by these TODOs in the future, it should be finalized in a dedicated change set and undergo a new security review before deployment.

Alleviation

[BOT Chain, 02/23/2026]: The team heeded the advice and resolved the issue by removing the TODO comments in commit 129e91c9cf9ab184cd12aed75e350c0ab299ba9d

BCC-07 | Inconsistency Between Comment And Code

Category	Severity	Location	Status
Inconsistency	Informational	BSCGovernor.sol (base): 44	Resolved

Description

In the contract `BSCGovernor` , the comment at the pointed location states `// starting propose requires totalSupply of GovBNB >= 10000000 * 1e18` , while the value is set to `PROPOSE_START_GOVBNB_SUPPLY_THRESHOLD = 10_000 ether;` .

Recommendation

We recommend making sure that the comment is consistent with the code implementation to avoid unexpected behaviors.

Alleviation

[BOT Chain, 02/23/2026]: The team heeded the advice and resolved the issue by fixing the comment in commit 129e91c9cf9ab184cd12aed75e350c0ab299ba9d

BCC-08 | Commented Out Code

Category	Severity	Location	Status
Coding Style	Informational	BSCValidatorSet.sol (base): 38, 79, 81~86, 81~86, 702~705, 714, 722	Resolved

Description

The highlighted lines contain commented-out code.

Recommendation

We recommend removing the commented-out code if it is not needed to improve the readability of the code.

Alleviation

[BOT Chain, 02/23/2026]: The team heeded the advice and resolved the issue by removing the commented-out codes in commit 129e91c9cf9ab184cd12aed75e350c0ab299ba9d

APPENDIX | BOT CHAIN - CHAIN

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.
Coding Issue	Coding Issue findings are about general code quality including, but not limited to, coding mistakes, compile errors, and performance issues.
Inconsistency	Inconsistency findings refer to different parts of code that are not consistent or code that does not behave according to its specification.
Volatile Code	Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases and may result in vulnerabilities.
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.

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.

Elevating Your Web3 Journey

Founded in 2017 by leading academics in the field of Computer Science from both Yale and Columbia University, CertiK is the largest 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.