CERTIK

Wadoozie

Security Assessment

CertiK Assessed on Apr 17th, 2026

CertiK Assessed on Apr 17th, 2026

Wadoozie

The security assessment was prepared by CertiK.

Executive Summary

TYPES	ECOSYSTEM	METHODS
ERC-20	EVM Compatible	Manual Review, Static Analysis

LANGUAGE	TIMELINE
Solidity	Preliminary comments published on 04/14/2026
	Final report published on 04/17/2026

Vulnerability Summary

5	4	0	1	0
Total Findings	Resolved	Partially Resolved	Acknowledged	Declined

1 Centralization	1 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.
1 Major	1 Resolved		Major risks may include logical errors that, under specific circumstances, could result in fund losses or loss of project control.
1 Medium	1 Resolved		Medium risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform.
1 Minor	1 Resolved		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.
1 Informational	1 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.

TABLE OF CONTENTS | WADOOZIE

CODEBASE | WADOOZIE

Repository

https://github.com/Wadoozie/SmartContracts

Commit

f3ede5f5680f118505e150ac793852d5c584da7b

a82cacb601f8f1b027a8f85659f86f816dcb047a

dc423af0df17d61229d4118f53d2245e51875d69

AUDIT SCOPE | WADOOZIE

Wadoozie/SmartContracts
contracts/Wadoozie.sol
contracts/Headquarters.sol
contracts/WadoozieTimelock.sol

APPROACH & METHODS | WADOOZIE

This audit was conducted for Wadoozie to evaluate the security and correctness of the smart contracts associated with the Wadoozie 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.

FINDINGS | WADOOZIE

5	0	1	1	1	1	1
Total Findings	Critical	Centralization	Major	Medium	Minor	Informational

This report has been prepared for Wadoozie to identify potential vulnerabilities and security issues within the reviewed codebase. During the course of the audit, a total of 5 issues were identified. Leveraging a combination of Manual Review & Static Analysis the following findings were uncovered:

ID	Title	Category	Severity	Status
WAD-01	**Initial Token Distribution**	**Centralization**	**Centralization**	**Acknowledged**
WAD-02	Queued Proposals Become Orphaned Across Timelock Migrations	Design Issue	Major	Resolved
WAD-03	Proposal Guardian Can Cause Governance Censorship And Permanent Deadlock	Governance	Medium	Resolved
WAD-04	Governance May Become Inactive Due To Missing Delegation Awareness	Governance	Minor	Resolved
WAD-05	Missing Late Quorum Protection May Enable Vote Sniping	Governance	Informational	Resolved

WAD-01 | Initial Token Distribution

Category	Severity	Location	Status
Centralization	**Centralization**	**contracts/Wadoozie.sol (base): 17**	**Acknowledged**

Description

All of the `WADZ` tokens are sent to the contract deployer or one or several externally-owned account (EOA) addresses. This is a centralization risk because the deployer or the owner(s) of the EOAs can distribute tokens without obtaining the consensus of the community. Any compromise to these addresses may allow a hacker to steal and sell tokens on the market, resulting in severe damage to the project.

Recommendation

It is recommended that the team be transparent regarding the initial token distribution process. The token distribution plan should be published in a public location that the community can access. The team should make efforts to restrict access to the private keys of the deployer account or EOAs. A multi-signature (⅔, ⅗) wallet can be used to prevent a single point of failure due to a private key compromise. Additionally, the team can lock up a portion of tokens, release them with a vesting schedule for long-term success, and deanonymize the project team with a third-party KYC provider to create greater accountability.

Alleviation

[Wadoozie, 04/16/2026]: **Acknowledged.** The total WADZ supply will be distributed across multiple multisig wallets upon token contract deployment. We will update this finding with the token distribution plan URL, multisig wallet addresses, and signer details once deployment is complete.

WAD-02 Queued Proposals Become Orphaned Across Timelock Migrations

Category	Severity	Location	Status
Design Issue	Major	contracts/Headquarters.sol (base): 110	Resolved

Description

`Headquarters` in `contracts/Headquarters.sol` uses OpenZeppelin's `GovernorTimelockControl` with a mutable timelock, but queued proposals are not bound to the specific `TimelockController` that scheduled them. When governance switches from `T1` to `T2` through `updateTimelock(TimelockController newTimelock)`, proposals already queued on `T1` are thereafter checked, executed, and canceled against `T2`. This desynchronizes governor state from the actual queued operation, causing already queued proposals to become misreported and unmanaged through the governor. In deployments where `EXECUTOR_ROLE` is granted to `address(0)`, stale batches on `T1` can still be executed directly if their payload does not rely on governor-only authorization.

Scenario

- *Shadow execution path:* proposal `P1` is queued on `T1`. Before `P1` executes, governance executes a migration proposal that calls `updateTimelock(T2)`. `Headquarters.state(P1)` now checks `T2` and no longer reflects the real queued operation on `T1`, while `Headquarters.execute(...)` also routes to `T2`. If `P1` does not rely on `onlyGovernance`, any user can call `T1.executeBatch(...)` after the delay and execute the stale batch directly.
- *Stranding path:* proposal `P2` is queued on `T1` and includes calls that require governor authorization on `Headquarters`. After migrating to `T2`, the governor cannot execute `P2` because it now targets `T2`, and direct execution from `T1` fails `_checkGovernance()` because `_executor()` resolves to `T2`. `P2` remains stranded until governance performs an explicit recovery action.

Recommendation

Bind each queued proposal to the timelock that scheduled it and use that stored timelock for all subsequent lifecycle actions, including state checks, execution, and cancellation. During `updateTimelock(...)`, prevent migration while proposals remain queued on the current timelock, or provide an explicit migration flow that re-queues or resolves all pending operations before switching. Also ensure the governor can accurately distinguish legacy queued operations from those on the active timelock so proposal status is not misreported. If direct execution from a retired timelock is not intended, close that path by avoiding open executor permissions on deprecated timelocks and fully decommissioning or revoking their roles before completing the migration.

Alleviation

[Wadoozie, 04/16/2026]: **Resolved.** Timelock migration has been permanently disabled by overriding updateTimelock() to unconditionally revert with HeadquartersTimelockImmutable().

The timelock is now immutably bound at construction, eliminating both the shadow execution and stranding scenarios described in this finding. A new governor deployment is required if a timelock change is ever needed.

The fix includes a test that proves even a fully passed and queued governance proposal attempting to call updateTimelock will revert on execution.

Commit: https://github.com/Wadoozie/SmartContracts/commit/6e9ecd67a02a72f8bd1609103f8f92e398ce76f7

[CertiK, 04/16/2026]: The team heeded the advice and resolved the issue in commit a82cacb601f8f1b027a8f85659f86f816dcb047a

WAD-03 Proposal Guardian Can Cause Governance Censorship And Permanent Deadlock

Category	Severity	Location	Status
Governance	Medium	contracts/Headquarters.sol (base): 44~62	Resolved

Description

The Headquarters contract integrates the GovernorProposalGuardian extension, introducing a privileged guardian role that is able to cancel any non-finalized proposal.

While the guardian is intended to act as an emergency safeguard, it introduces a centralization risk and potential governance failure mode. Specifically, the guardian retains unilateral authority to cancel proposals at any stage prior to execution, including proposals that attempt to modify or remove the guardian itself.

Although the contract enforces that the guardian can only be updated via governance (setProposalGuardian is onlyGovernance), this protection is insufficient. A malicious or compromised guardian can continuously cancel any proposal that attempts to revoke or replace its authority.

As a result, the governance system may enter a permanent deadlock state, where the guardian role becomes unremovable and effectively gains persistent veto power over all governance actions.

Scenario

- The guardian is set to an externally owned account (EOA) or a compromised multisig.
- A governance proposal is created to replace or remove the guardian.
- Before the proposal reaches execution, the guardian calls cancel() on the proposal.
- The community attempts to re-submit the proposal.
- The guardian continues canceling all such proposals.

As a result:

- The guardian becomes effectively permanent
- Governance loses the ability to self-correct
- All future proposals can be selectively censored by the guardian

Recommendation

- Use a secure multisig (e.g., 3/5 or higher) as the guardian instead of a single EOA.
- Consider implementing additional safeguards, such as:
  - Time-bound guardian privileges (e.g., guardian expires after a certain period)
  - Limited cancellation scope (e.g., restrict guardian to only cancel specific proposal types)

- Guardian override mechanisms, such as allowing proposals to bypass guardian cancellation under certain quorum thresholds

- Alternatively, evaluate whether the guardian role is necessary in production deployments, and consider removing it entirely to ensure full decentralization.

Alleviation

[Wadoozie, 04/16/2026]: **Resolved.** The GovernorProposalGuardian extension has been removed entirely from Headquarters, as recommended. There is no guardian role, no privileged cancel authority, and no admin of any kind. Cancellation now follows base Governor behavior only: a proposer may cancel their own proposal while it is still in the Pending state. Once voting begins, no one can unilaterally cancel. This eliminates the deadlock scenario completely and ensures

full decentralization with zero centralization risk.

Commit: https://github.com/Wadoozie/SmartContracts/commit/b75dea15e059b656e8455969fd70b65c6fd9bc1e

---

[CertiK, 04/16/2026]: The team heeded the advice and resolved the issue in commit a82cacb601f8f1b027a8f85659f86f816dcb047a

WAD-04 Governance May Become Inactive Due To Missing Delegation Awareness

Category	Severity	Location	Status
Governance	Minor	contracts/Headquarters.sol (base); contracts/Wadoozie.sol (base)	Resolved

Description

The governance system relies on ERC20Votes for voting power tracking. However, voting power in ERC20Votes is not automatically assigned based on token balances. Token holders must explicitly delegate their voting power (including self-delegation) in order to participate in governance.

At the same time, quorum is calculated as a percentage of the total token supply via GovernorVotesQuorumFraction. This creates a mismatch where:

- Total supply is fully counted toward quorum
- But only delegated tokens are counted toward voting power

If a significant portion of token holders do not delegate their votes, proposals may consistently fail to reach quorum, effectively rendering governance inactive.

Recommendation

Encourage or enforce delegation by:

- Prompting users to self-delegate upon receiving tokens
- Providing UI-level guidance for delegation

Consider:

- Lowering quorum requirements to reflect realistically delegated supply
- Implementing mechanisms that track participation rates before setting quorum
- Clearly document the requirement for delegation in user-facing materials

Alleviation

[Wadoozie, 04/16/2026]: **Acknowledged.** This is an inherent design property of OpenZeppelin's ERC20Votes, not a vulnerability opt-in delegation is the standard across all Governor-based governance systems. Our quorum is set at 4% to account for partial delegation. The website will include a DAO section on the home page, a dedicated DAO page with an easy delegation flow, and a custom documentation page explaining the delegation requirement and governance process.

[Wadoozie, 04/17/2026]:
https://github.com/Wadoozie/SmartContracts/commit/dc423af0df17d61229d4118f53d2245e51875d69

Please check if this would resolve the issue - if it does then we will merge this in main.

[CertiK, 04/17/2026]: The team heeded the advice and resolved the issue in commit dc423af0df17d61229d4118f53d2245e51875d69

WAD-05 | Missing Late Quorum Protection May Enable Vote Sniping

Category	Severity	Location	Status
Governance	Informational	contracts/Headquarters.sol (base)	Resolved

Description

The Headquarters contract does not include the GovernorPreventLateQuorum extension, which is designed to mitigate vote sniping risks.

Without late quorum protection, it is possible for a proposal to reach quorum near the end of the voting period, leaving insufficient time for other participants to react or cast opposing votes.

This creates an advantage for large token holders, who can strategically delay their votes until the final blocks to push proposals over the quorum threshold.

Recommendation

Integrate GovernorPreventLateQuorum to automatically extend voting when quorum is reached late

Alternatively:

- Increase voting period duration
- Monitor voting participation patterns before setting governance parameters

Alleviation

[Wadoozie, 04/16/2026]: **Resolved.** Integrated OpenZeppelin's GovernorPreventLateQuorum extension. When quorum is reached near the end of the voting period, the deadline automatically extends to guarantee a minimum of 7200 blocks (~1 day) of

remaining voting time. This prevents large token holders from sniping votes at the last moment. The vote extension parameter is configurable via governance (setLateQuorumVoteExtension is onlyGovernance).

Commit: https://github.com/Wadoozie/SmartContracts/commit/a82cacb601f8f1b027a8f85659f86f816dcb047a

---

[CertiK, 04/16/2026]: The team heeded the advice and resolved the issue in commit a82cacb601f8f1b027a8f85659f86f816dcb047a. We would like to suggest the team making sure setting a meaningful non-zero value for _voteExtension when deployment

APPENDIX | WADOOZIE

Finding Categories

Categories	Description
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.
Governance	Governance findings indicate issues related to the management of the code.

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.

Elevate Your Web3 Journey

CertiK is the largest Web3 security platform combining formal verification with audits and comprehensive security solutions.