SecureDApp

13th Aug 2024

SecureDapp
235, 2nd & 3rd Floor, 13th Cross Rd, Indira Nagar II Stage,
Hoysala Nagar, Indiranagar, Bengaluru, Karnataka 560038
hello@securedapp.in

Executive Summary	
AUDIT_HASH	581747e62520a7ead813639e8165fd2a9a0082a5c2b673baff1102ad47a000e8
Contracts	10
Lines	1056
Assembly Lines	0
ERCs	ERC20, ERC2612, ERC165

Audit Findings	Count
CRITICAL	1
MEDIUM	8
LOW	2
INFORMATIONAL	26
OPTIMIZATIONS	0

Vulnerability Found	Critical Level
Incorrect exp	High

Description

Incorrect exponentiation poses a medium risk as it can lead to incorrect mathematical calculations, potentially affecting the accuracy or integrity of computations within the contract.

Recommended Solution

Review and correct exponentiation operations to ensure accurate mathematical calculations. Use standard libraries or built-in functions for exponentiation to minimize the risk of errors and ensure computational integrity.

Vulnerability Found	Critical Level
Divide before multiply	Medium

Description

Imprecise arithmetic operations order poses a medium risk as it can lead to incorrect mathematical calculations, potentially affecting the accuracy or integrity of computations within the contract.

Vulnerability Found	Critical Level
Shadowing local	Low

Description

Local variables shadowing poses a low risk as it can lead to confusion or unintended consequences due to ambiguity in variable references or function calls.

Recommended Solution

Prevent local variables from shadowing to avoid confusion or unintended consequences. Use unique names for local variables to maintain clarity and readability in the codebase.

Vulnerability Found	Critical Level
Timestamp	Low

Description

Dangerous usage of block.timestamp poses a low risk as it can lead to issues with security or unexpected behavior related to timestamp-dependent logic.

Vulnerability Found	Critical Level
Assembly	Informational

Description

Assembly usage is categorized as informational as it provides insights into the usage of assembly code within the contract, which can be used for optimization or fine-tuning contract performance.

Recommended Solution

Eliminate the use of assembly code where possible to improve code readability, reduce complexity, and optimize gas usage. Refactor code to use higher-level Solidity constructs whenever possible to maintain consistency and readability.

Vulnerability Found	Critical Level
Dead code	Informational

Description

Functions that are not used is categorized as informational as it provides insights into unused functions within the contract, which can aid in code review or optimization efforts.

Recommended Solution

Remove unused functions and code segments to improve code readability, reduce complexity, and minimize attack surface. Conduct regular code reviews and refactorings to identify and eliminate redundant or obsolete code.

Vulnerability Found	Critical Level
Solc version	Informational

Description

Incorrect Solidity version is categorized as informational as it provides insights into the usage of incorrect Solidity versions within the contract, which can aid in ensuring compatibility or adherence to best practices.

Recommended Solution

Update the Solidity version to the correct one specified for the contract. Ensure that the contract is compatible with the targeted Solidity compiler version to avoid potential issues or unexpected behavior.

Vulnerability Found	Critical Level
Naming convention	Informational

Description

Conformity to Solidity naming conventions is categorized as informational as it provides insights into adherence to Solidity naming conventions within the contract, which can aid in code readability or maintainability.

Recommended Solution

Enforce Solidity naming conventions consistently throughout the contract codebase. Use descriptive and meaningful names for variables, functions, and contracts to enhance code readability and maintainability.

Vulnerability Found	Critical Level
Too many digits	Informational

Description

Conformance to numeric notation best practices is categorized as informational as it provides insights into adherence to numeric notation best practices within the contract, which can aid in code readability or maintainability.

Recommended Solution

Ensure adherence to numeric notation best practices by avoiding excessive digits in numerical values. Trim down the number of digits to maintain code readability and conform to standard practices, enhancing maintainability and reducing the likelihood of errors.

Disclaimer

Topic	Description
Purpose	This audit report is provided for informational purposes only
Scope	The audit was performed based on the state of the software at the time of the audit and may not reflect its current state or any subsequent changes.
Limitations	While every effort has been made to ensure the accuracy and completeness of this report, no guarantee is made that all vulnerabilities or issues have been identified. Security audits do not guarantee complete system security.
Recommendations	The recommendations provided in this report are based on the best judgment of SecureDApp's security professionals. Implementation of these recommendations is at the discretion of the software's maintainers.
Responsibility	It remains the responsibility of the software's maintainers and users to ensure its security and proper functionality. SecureDApp does not accept any liability for any damage or loss caused due to overlooked vulnerabilities or misinterpretations in this report.

Contact Us

Email	hello@securedapp.in
Phone	9606015868
Address	SecureDApp Solutions Pvt. Ltd. 235, 2nd & 3rd Floor,13th Cross Rd, Indira Nagar II Stage,Hoysala Nagar, Indiranagar, Bengaluru, Karnataka 560038
Website	securedapp.io
Business Hours	Monday to Friday, 9 AM - 6 PM IST