CFG NINJA

SECURITY ASSESSMENT

DIGIVERSE Token

December 5, 2023

Audit Status: Pass

Digiverse
DIGITAL UNIVERSE EXPERIENCE

BLADE POOL

Risk Analysis

Classifications of Risk Results | Manual

Classification	Description
Critical	Danger or Potential Problems.
High	Be Careful or Fail test.
Low	Pass, Not-Detected or Safe Item.
Informational	Function Detected

Review Risk Results | DIGIVERSE.

Contract Security	Description
Buy Tax	3%
Sale Tax	3%
Cannot Buy	Pass
Cannot Sale	Pass
Max Tax	3%
Modify Tax	No
Fee Check	Pass
Is Honeypot?	Not Detected
Trading Cooldown	Not Detected
Enable Trade?	Pass
Pause Transfer?	Not Detected

Contract Security	Description
Max Tx?	Pass
Is Anti Whale?	Not Detected
Is Anti Bot?	Not Detected
Is Blacklist?	Not Detected
Blacklist Check	Pass
is Whitelist?	Not Detected
Can Mint?	Pass
Is Proxy?	Not Detected
Can Take Ownership?	Not Detected
Hidden Owner?	Not Detected
Owner	0xf0B04a47846aAFb8aCcb050bcd5F7a5725596afe
Self Destruct?	Not Detected
External Call?	Not Detected
Other?	Not Detected
Holders	1
Audit Confidence	Medium

The summary section reveals the strengths and weaknesses identified during the assessment, including any vulnerabilities or potential risks that may exist. It serves as a valuable snapshot of the overall security status of the audited project. However, it is highly recommended to read the entire security assessment report for a comprehensive understanding of the findings. The full report provides detailed insights into the assessment process, methodology, and specific recommendations for addressing the identified issues.

Project Overview | Token Summary

Parameter	Result
Address	0x60d2f1042dc74Aa8cE5904307063A52fd9Fa3d31
Name	DIGIVERSE
Token Tracker	DIGIVERSE (XDIGI)
Decimals	18
Supply	100,000,000
Platform	Binance Smart Chain
compiler	v0.8.19+commit.7dd6d404
Contract Name	DIGIVERSE
Optimization	Yes with 200 runs
LicenseType	MIT
Language	Solidity
Codebase	https://bscscan.com/token/0x60d2f1042dc74Aa8cE5904307063A52fd9Fa3d31#code
Payment Tx	Corporate

Project Overview | Simulation Summary

Parameter	Result
Transfer From Owner	Pass
Transfer From Holder	Pass
Add Liquidity	Pass
RemoveLiquidity	Pass
Buy from Owner	Pass
Buy from Holder	Pass
Sale from Owner	Pass
Sale from Holder	Pass
Remove Liquidity	Pass
SwapAndLiquify	Pass
SwapAndSale w/Fee	Pass
SwapAndSale TX	https://testnet.bscscan.com/tx/0xf055e5c3408249c31ef26251814de47481a0b0c3a67b851c261459073bd0ec11
SwapAndSaleNoFee	Pass
SwapAndSale No/Fee TX	
ExcludeFromFees	Pass
LaunchPad	PinkSale
Pool Creation	Pass
Pool Creation TX	

Parameter	Result
Pool Finalize	Pass
Pool Finalize TX	
Enable	Pass

The following quick summary it's added to the project overview; however, there are more details about the audit and its results. Please read every detail.

Project Overview | Main Contract Assessed

Name	Contract	Live
DIGIVERSE	0x60d2f1042dc74Aa8cE5904307063A52fd9Fa3d31	Yes

Project Overview | TestNet Contract Assessed

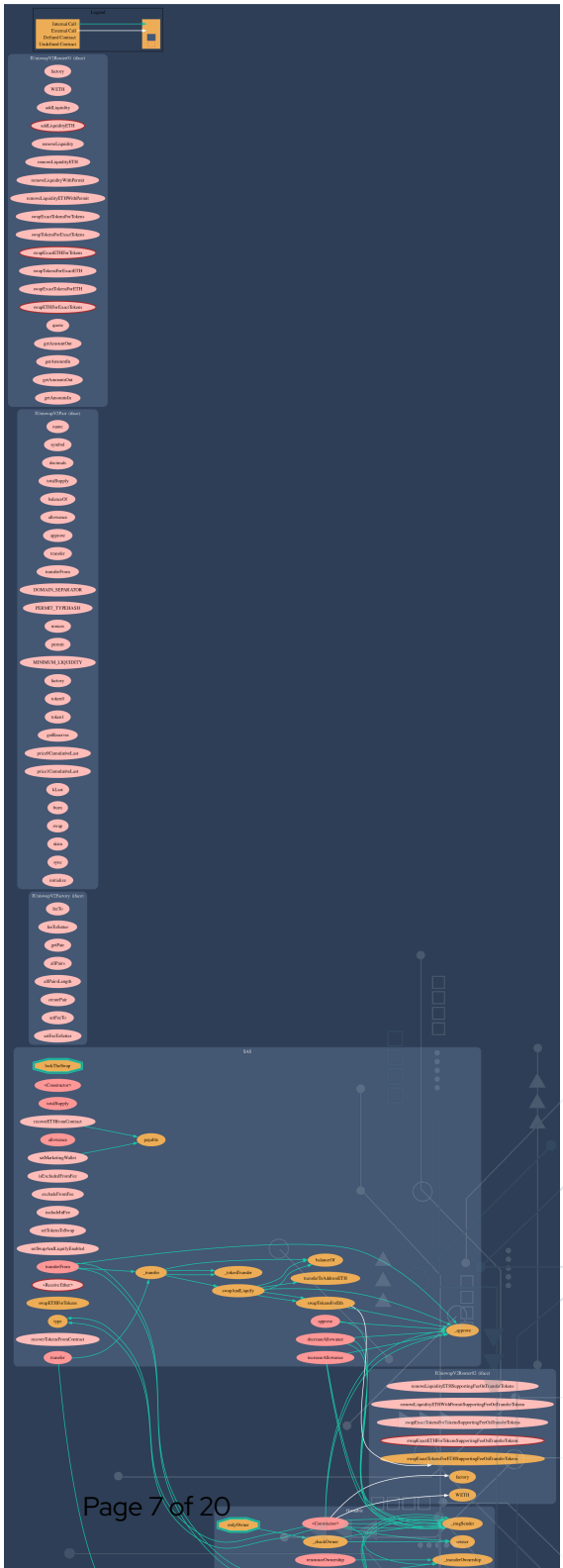
Name	Contract	Live
DIGIVERSE	0x122186f37E35CED4bdEce776E23A35a5910A1450	Yes

Project Overview | Solidity Code Provided

SolID	File Sha-1	FileName
DIGI	d9c47a0506a5811d13a210abe8d7a526268ee0c1	Digiverse1.sol

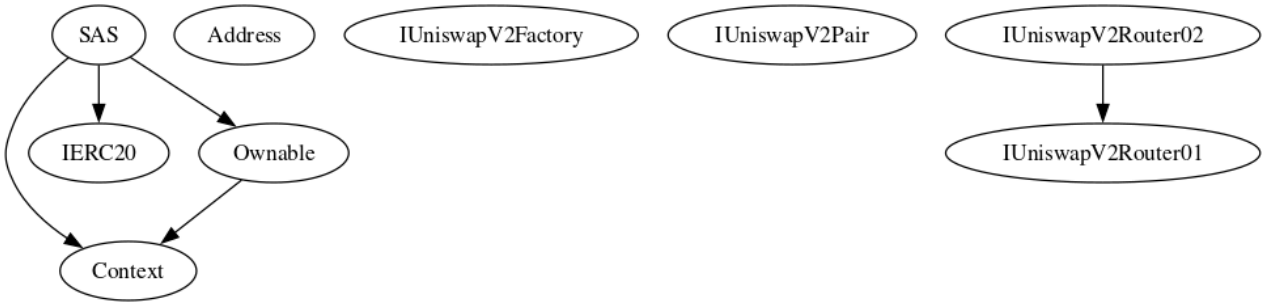
Project Overview | Call Graph

The Smart Contract Graph is a visual representation of the interconnectedness and relationships between smart contracts within a blockchain network. It provides a comprehensive view of the interactions and dependencies between different smart contracts, allowing developers and users to analyze and understand the flow of data and transactions within the network. The Smart Contract Graph enables better transparency, security, and efficiency in decentralized applications by facilitating the identification of potential vulnerabilities, optimizing contract execution, and enhancing overall network performance.



Project Overview | Inheritance.

Smart contract inheritance is a concept in blockchain programming where one smart contract can inherit properties and functionalities from another existing smart contract. This allows for code reuse and modularity, making the development process more efficient and scalable. Inheritance enables the child contract to access and utilize the variables, functions, and modifiers defined in the parent contract, thereby inheriting its behavior and characteristics. This feature is particularly useful in complex decentralized applications (dApps) where multiple contracts need to interact and share common functionalities. By leveraging smart contract inheritance, developers can create more organized and maintainable code structures, promoting code reusability and reducing redundancy.



Project Overview | Smart Contract Vulnerability Checks

The Smart Contract Weakness Classification Registry (SWC Registry) is an implementation of the weakness classification scheme proposed in EIP-1470. It is loosely aligned to the terminologies and structure used in the Common Weakness Enumeration (CWE) while overlaying a wide range of weakness variants that are specific to smart contracts.

ID	Severity	Name	File	location
SWC-100	Pass	Function Default Visibility	Digiverse1.sol	L: 0 C: 0
SWC-101	Pass	Integer Overflow and Underflow.	Digiverse1.sol	L: 0 C: 0
SWC-102	Pass	Outdated Compiler Version file.	Digiverse1.sol	L: 0 C: 0
SWC-103	Pass	A floating pragma is set.	Digiverse1.sol	L: 0 C: 0
SWC-104	Pass	Unchecked Call Return Value.	Digiverse1.sol	L: 0 C: 0
SWC-105	Pass	Unprotected Ether Withdrawal.	Digiverse1.sol	L: 0 C: 0
SWC-106	Pass	Unprotected SELFDESTRUCT Instruction	Digiverse1.sol	L: 0 C: 0
SWC-107	Pass	Read of persistent state following external call.	Digiverse1.sol	L: 0 C: 0
SWC-108	Pass	State variable visibility is not set..	Digiverse1.sol	L: 0 C: 0
SWC-109	Pass	Uninitialized Storage Pointer.	Digiverse1.sol	L: 0 C: 0
SWC-110	Pass	Assert Violation.	Digiverse1.sol	L: 0 C: 0
SWC-111	Pass	Use of Deprecated Solidity Functions.	Digiverse1.sol	L: 0 C: 0
SWC-112	Pass	Delegate Call to Untrusted Callee.	Digiverse1.sol	L: 0 C: 0
SWC-113	Pass	Multiple calls are executed in the same transaction.	Digiverse1.sol	L: 0 C: 0
SWC-114	Pass	Transaction Order Dependence.	Digiverse1.sol	L: 0 C: 0

ID	Severity	Name	File	location
SWC-115	Pass	Authorization through tx.origin.	Digiverse1.sol	L: 0 C: 0
SWC-116	Pass	A control flow decision is made based on The block.timestamp environment variable.	Digiverse1.sol	L: 0 C: 0
SWC-117	Pass	Signature Malleability.	Digiverse1.sol	L: 0 C: 0
SWC-118	Pass	Incorrect Constructor Name.	Digiverse1.sol	L: 0 C: 0
SWC-119	Pass	Shadowing State Variables.	Digiverse1.sol	L: 0 C: 0
SWC-120	Pass	Potential use of block.number as source of randonmness.	Digiverse1.sol	L: 0 C: 0
SWC-121	Pass	Missing Protection against Signature Replay Attacks.	Digiverse1.sol	L: 0 C: 0
SWC-122	Pass	Lack of Proper Signature Verification.	Digiverse1.sol	L: 0 C: 0
SWC-123	Pass	Requirement Violation.	Digiverse1.sol	L: 0 C: 0
SWC-124	Pass	Write to Arbitrary Storage Location.	Digiverse1.sol	L: 0 C: 0
SWC-125	Pass	Incorrect Inheritance Order.	Digiverse1.sol	L: 0 C: 0
SWC-126	Pass	Insufficient Gas Griefing.	Digiverse1.sol	L: 0 C: 0
SWC-127	Pass	Arbitrary Jump with Function Type Variable.	Digiverse1.sol	L: 0 C: 0
SWC-128	Pass	DoS With Block Gas Limit.	Digiverse1.sol	L: 0 C: 0
SWC-129	Pass	Typographical Error.	Digiverse1.sol	L: 0 C: 0
SWC-130	Pass	Right-To-Left-Override control character (U+202E).	Digiverse1.sol	L: 0 C: 0
SWC-131	Pass	Presence of unused variables.	Digiverse1.sol	L: 0 C: 0

ID	Severity	Name	File	location
SWC-132	Pass	Unexpected Ether balance.	Digiverse1.sol	L: 0 C: 0
SWC-133	Pass	Hash Collisions with Multiple Variable Length Arguments.	Digiverse1.sol	L: 0 C: 0
SWC-134	Pass	Message call with hardcoded gas amount.	Digiverse1.sol	L: 0 C: 0
SWC-135	Pass	Code With No Effects (Irrelevant/Dead Code).	Digiverse1.sol	L: 0 C: 0
SWC-136	Pass	Unencrypted Private Data On-Chain.	Digiverse1.sol	L: 0 C: 0

We scan the contract for additional security issues using MYTHX and industry-standard security scanning tools.

Technical Findings | Overview.

Smart contract security audits classify risks into several categories: Critical, High, Medium, Low, and Informational. These classifications help assess the severity and potential impact of vulnerabilities found in smart contracts.

Classification of Risk

Severity	Description
Critical	Critical risks are the most severe and can have a significant impact on the smart contracts functionality, security, or the entire system. These vulnerabilities can lead to the loss of user funds, unauthorized access, or complete system compromise.
High	High-risk vulnerabilities have the potential to cause significant harm to the smart contract or the system. While not as severe as critical risks, they can still result in financial losses, data breaches, or denial of service attacks.
Medium	Medium-risk vulnerabilities pose a moderate level of risk to the smart contracts security and functionality. They may not have an immediate and severe impact but can still lead to potential issues if exploited. These risks should be addressed to ensure the contracts overall security.
Low	Low-risk vulnerabilities have a minimal impact on the smart contracts security and functionality. They may not pose a significant threat, but it is still advisable to address them to maintain a robust security posture.
Informational	Informational risks are not actual vulnerabilities but provide useful information about potential improvements or best practices. These findings may include suggestions for code optimizations, documentation enhancements, or other non-critical areas for improvement.

By categorizing risks into these classifications, smart contract security audits can prioritize the resolution of critical and high-risk vulnerabilities to ensure the contract's overall security and protect user funds and data.

Category	Severity	Location	Status
Coding Style	Medium	Digiverse1.sol: L: 0 C: 0	Detected

Description

In a smart contract, the concept of "onlyOwner" functions refers to certain functions that can only be executed by the owner or creator of the contract. These functions are typically designed to perform critical actions or modify sensitive data within the contract. By restricting access to these functions, the contract owner maintains control and ensures the integrity and security of the contract.

Function Name	Parameters	Visibility
renounceOwnership		Public
transferOwnership	address newOwner	Public
excludeFromFee	address account	external
includeInFee	address account	external
setTokensToSwap	uint256 _minimumTokensBeforeSwap	external
setSwapAndLiquifyEnabled	bool _enabled	external
setMarketingWallet	address _marketingWallet	external

Recommendation

Inheriting from Ownable and calling its constructor on yours ensures that the address deploying your contract is registered as the owner. The onlyOwner modifier makes a function revert if not called by the address registered as the owner. It is important that

deployr or owner secure the credentials that has owner priviledge to ensure the security of the project.

Mitigation

References:

Guide to Ownership and Access Control in Solidity

Writing Clean Code for Solidity: Best Practices for Solidity Development

Technical Findings | Findings.

In this document, we present the findings and results of the smart contract security audit. The identified vulnerabilities, weaknesses, and potential risks are outlined, along with recommendations for mitigating these issues. It is crucial for the team to address these findings promptly to enhance the security and trustworthiness of the smart contract code.

Severity	Found	Pending	Resolved
Critical	0	0	0
High	0	0	0
Medium	1	0	0
Low	0	0	0
Informational	0	0	0
Total	1	0	0

In a smart contract, a technical finding summary refers to a compilation of identified issues or vulnerabilities discovered during a security audit. These findings can range from coding errors and logical flaws to potential security risks. It is crucial for the project owner to thoroughly review each identified item and take necessary actions to resolve them. By carefully examining the technical finding summary, the project owner can gain insights into the weaknesses or potential threats present in the smart contract. They should prioritize addressing these issues promptly to mitigate any risks associated with the contract's security. Neglecting to address any identified item in the security audit can expose the smart contract to significant risks. Unresolved vulnerabilities can be exploited by malicious actors, potentially leading to financial losses, data breaches, or other detrimental consequences. To ensure the integrity and security of the smart contract, the project owner should engage in a comprehensive review process. This involves understanding the nature and severity of each identified item, consulting with experts if needed, and implementing appropriate fixes or enhancements. Regularly updating and maintaining the smart contract's codebase is also essential to address any emerging security concerns. By diligently reviewing and resolving all identified items in the technical finding summary, the project owner can significantly reduce the risks associated with the smart contract and enhance its overall security posture.

Project Overview | Social Media Checks for XDIGI.

Social Media	URL	Result
Website	https://digiversecrypto.com/	Pass
Telegram	https://t.me/Digiversee	Pass
Twitter	https://twitter.com/Digiversede/	Pass
Facebook	https://www.facebook.com/digiversede360	Pass
Reddit	https://www.reddit.com/user/DigiMetaVerse/	Pass
Instagram	https://www.instagram.com/digiversede/	Pass
CoinGecko	N/A	N/A
Github	https://github.com/DigiverseProject	Pass
CMC	N/A	N/A
Other	https://medium.com/@digiversee	Pass

From a security assessment standpoint, inspecting a project's social media presence is essential. It enables the evaluation of the project's reputation, credibility, and trustworthiness within the community. By analyzing the content shared, engagement levels, and the response to any security-related incidents, one can assess the project's commitment to security practices and its ability to handle potential threats.

Social Media Information Notes:

Auditor Notes: Website need improvements.

Project Owner Notes:

Assessment Results | Score Results for XDIGI.

Review	Score
Overall Score	99/100
Auditor Score	90/100

Review by Section	Score
Manual Scan Score	26
SWC Scan Score	37
Advance Check Score	36

Our security assessment or audit score system for the smart contract and project follows a comprehensive evaluation process to ensure the highest level of security. The system assigns a score based on various security parameters and benchmarks, with a passing score set at 80 out of a total attainable score of 100.The assessment process includes a thorough review of the smart contracts codebase, architecture, and design principles. It examines potential vulnerabilities, such as code bugs, logical flaws, and potential attack vectors. The evaluation also considers the adherence to best practices and industry standards for secure coding. Additionally, the system assesses the projects overall security measures, including infrastructure security, data protection, and access controls. It evaluates the implementation of encryption, authentication mechanisms, and secure communication protocols. To achieve a passing score, the smart contract and project must attain a minimum of 80 points out of the total attainable score of 100. This ensures that the system has undergone a rigorous security assessment and meets the required standards for secure operation.

Audit Passed

- No issues or vulnerabilities were found.
- This contract has 2% for marketing and 1% for liquidity.

Auditor Score =90
Audit Passed

Finding Categories

Centralization / Privilege

Centralization / Privilege findings refer to either feature logic or implementation of components that actagainst the nature of decentralization, such as explicit ownership or specialized access roles incombination with a mechanism to relocate funds.

Gas Optimization

Gas Optimization findings do not affect the functionality of the code but generate different, more optimalEVM 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 howblock.timestamp works.

Control Flow

Control Flow findings concern the access control imposed on functions, such as owner-only functionsbeing 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 mayresult in a vulnerability.

Coding Style

Coding Style findings usually do not affect the generated byte-code but rather comment on how to makethe codebase more legible and, as a result, easily maintainable.

Inconsistency

Inconsistency findings refer to functions that should seemingly behave similarly yet contain different code,such as a constructor assignment imposing different require statements on the input variables than a setterfunction.

Coding Best Practices

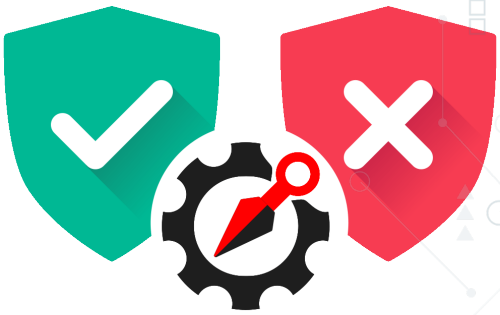
ERC 20 Conding Standards are a set of rules that each developer should follow to ensure the code meet a set of creterias and is readable by all the developers.

Assessment | Disclaimer.

The purpose of this disclaimer is to outline the responsibilities and limitations of the security assessment and smart contract audit conducted by Bladepool/CFG NINJA. By engaging our services, the project owner acknowledges and agrees to the following terms:

1. Limitation of Liability: Bladepool/CFG NINJA shall not be held liable for any damages, losses, or expenses incurred as a result of any contract malfunctions, vulnerabilities, or exploits discovered during the security assessment and smart contract audit. The project owner assumes full responsibility for any consequences arising from the use or implementation of the audited smart contract. 2. No Guarantee of Absolute Security: While Bladepool/CFG NINJA employs industry-standard practices and methodologies to identify potential security risks, it is important to note that no security assessment or smart contract audit can provide an absolute guarantee of security. The project owner acknowledges that there may still be unknown vulnerabilities or risks that are beyond the scope of our assessment. 3. Transfer of Responsibility: By engaging our services, the project owner agrees to assume full responsibility for addressing and mitigating any identified vulnerabilities or risks discovered during the security assessment and smart contract audit. It is the project owner s sole responsibility to ensure the proper implementation of necessary security measures and to address any identified issues promptly. 4. Compliance with Applicable Laws and Regulations: The project owner acknowledges and agrees to comply with all applicable laws, regulations, and industry standards related to the use and implementation of smart contracts. Bladepool/CFG NINJA shall not be held responsible for any non-compliance by the project owner. 5. Third-Party Services: The security assessment and smart contract audit conducted by Bladepool/CFG NINJA may involve the use of third-party tools, services, or technologies. While we exercise due diligence in selecting and utilizing these resources, we cannot be held liable for any issues or damages arising from the use of such third-party services. 6. Confidentiality: Bladepool/CFG NINJA maintains strict confidentiality regarding all information and data obtained during the security assessment and smart contract audit. However, we cannot guarantee the security of data transmitted over the internet or through any other means. 7. Not a Financial Advice: Bladepool/CFG NINJA please note that the information provided in the security assessment or audit should not be considered as financial advice. It is always recommended to consult with a financial professional or do thorough research before making any investment decisions.

By engaging our services, the project owner acknowledges and accepts these terms and releases Bladepool/CFG NINJA from any liability, claims, or damages arising from the security assessment and smart contract audit. It is recommended that the project owner consult legal counsel before entering into any agreement or contract.



CFG NINJA AUDITS