



DESSERT
FINANCE

Nemesis Downfall (NMSIS)

MultiChain Audit

Performed at block **16515462**

PERFORMED BY DESSERT FINANCE

FOR ETH CONTRACT ADDRESS: `0x7B6BbbeAC6a7F5681eC8e250B9Aeb45a42Bdc2Cf`
FOR POLYGON CONTRACT ADDRESS: `0x3F5B2063E500492463589b17F9Ea9cf2aF9771cF`

INITIAL DISCLAIMER

Dessert Finance provides due-diligence project audits for various projects. Dessert Finance in no way guarantees that a project will not remove liquidity, sell off team supply, or otherwise exit scam.

Dessert Finance does the legwork and provides public information about the project in an easy-to-understand format for the common person.

Agreeing to an audit in no way guarantees that a team will not remove *all* liquidity (“Rug Pull”), remove liquidity slowly, sell off tokens, quit the project, or completely exit scam. There is also no way to prevent private sale holders from selling off their tokens. It is ultimately your responsibility to read through all documentation, social media posts, and contract code of each individual project to draw your own conclusions and set your own risk tolerance.

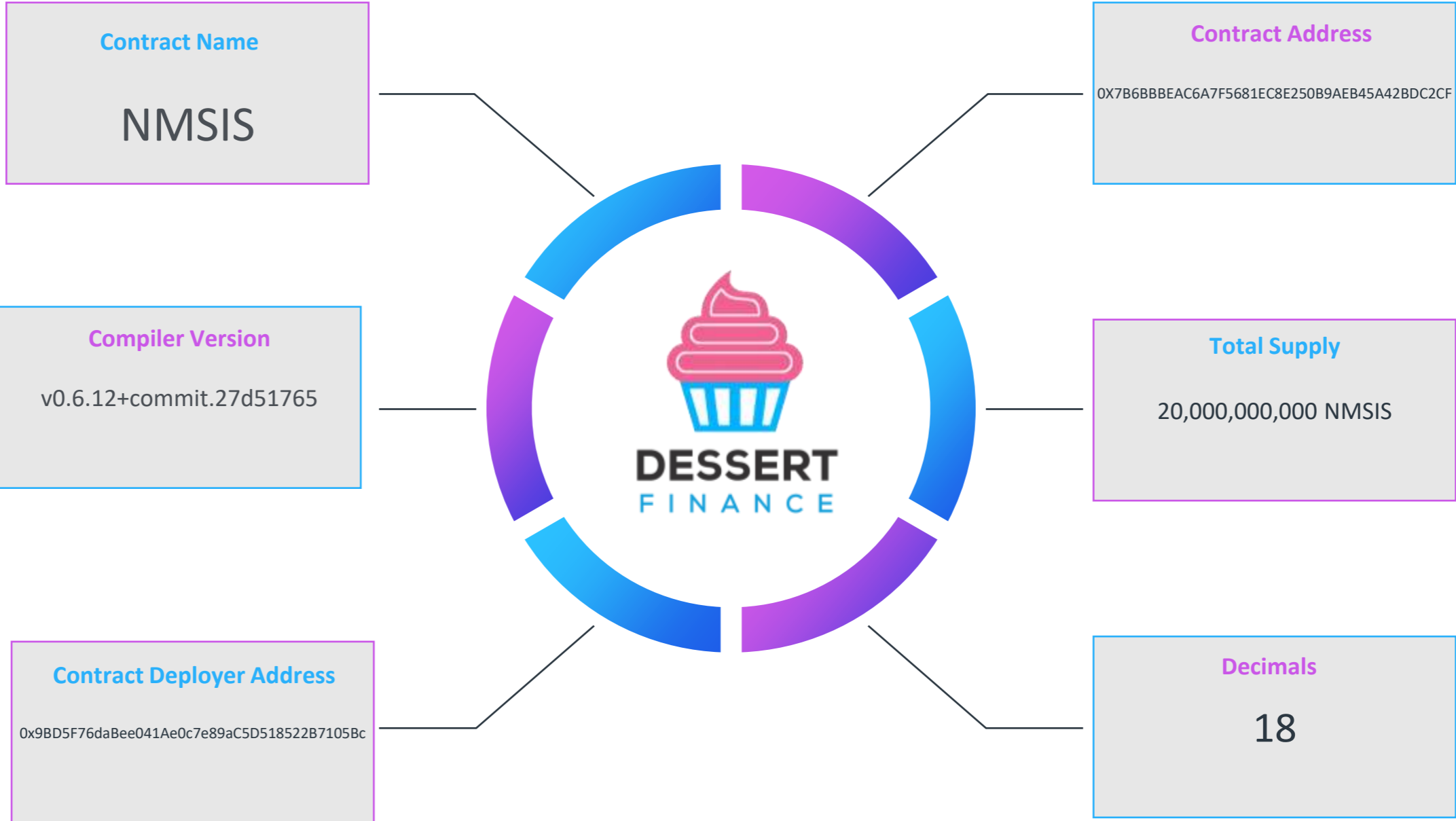
Dessert Finance in no way takes responsibility for any losses, nor does Dessert Finance encourage any speculative investments. The information provided in this audit is for information purposes only and should not be considered investment advice. Dessert Finance does not endorse, recommend, support, or suggest any projects that have been audited. An audit is an informational report based on our findings, We recommend you do your own research, we will never endorse any project to invest in.

Table of Contents



1. Contract Code Audit – Token Overview
2. ERC-20 Contract Code Audit – Overview
3. ERC-20 Contract Code Audit – Vulnerabilities Checked
4. Contract Code Audit – Contract Ownership
5. Contract Code Audit – Owner Accessible Functions
6. Liquidity Ownership – Locked / Unlocked
7. Contract Code Audit – Mint Functions
8. Contract Transaction Fees
9. Website Overview
10. Social Media
11. Top Token Holders/Wallets
12. Review of Team
13. Business Registration
14. Roadmap
15. Disclaimers

Contract Code Audit – Token Overview



ERC-20 Contract Code Audit – Overview

Dessert Finance was commissioned to perform an audit on Nemesis Downfall (NMSIS)

```

*Submitted for verification at Etherscan.io on 2021-12-12
*/
// SPDX-License-Identifier: MIT
pragma solidity 0.6.12;

interface IERC20 {
    /**
     * @dev Returns the amount of tokens in existence.
     */
    function totalSupply() external view returns (uint256);

    /**
     * @dev Returns the token decimals.
     */
    function decimals() external view returns (uint8);

    /**
     * @dev Returns the token symbol.
     */
    function symbol() external view returns (string memory);

    /**
     * @dev Returns the token name.
     */
    function name() external view returns (string memory);

    /**
     * @dev Returns the top token owner.
     */
    function getOwner() external view returns (address);

    /**
     * @dev Returns the amount of tokens owned by 'account'.
     */
    function balanceOf(address account) external view returns (uint256);

    /**
     * @dev Moves 'amount' tokens from the caller's account to 'recipient'.
     * Returns a boolean value indicating whether the operation succeeded.
     * Emits a [Transfer] event.
     */
    function transfer(address recipient, uint256 amount) external returns (bool);

    /**
     * @dev Returns the remaining number of tokens that 'spender' will be
     * allowed to spend on behalf of 'owner' through (transferFrom). This is
     * zero by default.
     * This value changes when [approve] or [transferFrom] are called.
     */
    function allowance(address owner, address spender) external view returns (uint256);

    /**
     * @dev Sets 'amount' as the allowance of 'spender' over the caller's tokens.
     * Returns a boolean value indicating whether the operation succeeded.
     *
     * WARNING: Ensure that changing an allowance with this method brings the risk
     * that someone may use both the old and the new allowance by unfortunate
     * transaction ordering. One possible solution to mitigate this race
     * condition is to first reduce the spender's allowance to 0 and set the
     * desired value afterwards.
     * https://github.com/ethereum/EIPs/issues/20#issuecomment-26352779
     */
    function approve(address spender, uint256 amount) external returns (bool);
}

```

Contract Address

0x7B6BbbeAC6a7F5681eC8e250B9Aeb45a42Bdc2Cf

TokenTracker

Nemesis Downfall (NMSIS)

Contract Creator

0x9bd5f76dabee041ae0c7e89ac5d518522b7105bc

Source Code

Contract Source Code Verified

Contract Name

NMSIS

Other Settings

default evmVersion, MIT

Compiler Version

v0.6.12+commit.27d51765

Optimization Enabled

No with 200 runs

Code is truncated to fit the constraints of this document.

[The code in its entirety can be viewed here.](#)

The contract code is **verified** on Etherscan.

ERC-20 Contract Code Audit – Vulnerabilities Checked

Vulnerability Tested	AI Scan	Human Review	Result
Compiler Errors	Complete	Complete	✓ Low / No Risk
Outdated Compiler Version	Complete	Complete	✓ Low / No Risk
Integer Overflow	Complete	Complete	✓ Low / No Risk
Integer Underflow	Complete	Complete	✓ Low / No Risk
Correct Token Standards Implementation	Complete	Complete	✓ Low / No Risk
Timestamp Dependency for Crucial Functions	Complete	Complete	✓ Low / No Risk
Exposed _Transfer Function	Complete	Complete	✓ Low / No Risk
Transaction-Ordering Dependency	Complete	Complete	✓ Low / No Risk
Unchecked Call Return Variable	Complete	Complete	✓ Low / No Risk
Use of Deprecated Functions	Complete	Complete	✓ Low / No Risk
Unprotected SELFDESTRUCT Instruction	Complete	Complete	✓ Low / No Risk
State Variable Default Visibility	Complete	Complete	✓ Low / No Risk
Deployer Can Access User Funds	Complete	Complete	✓ Low / No Risk

The contract code is **verified** on EtherScan.

The vulnerabilities listed above were not found in the token's Smart Contract.

Contract Code Audit – Contract Ownership

Contract Ownership has not been renounced at the time of Audit



The contract ownership is not currently renounced.

We have placed the contract owner address below for your viewing:

[0x0d75e0d766bafc34297fbbae8c13caea6db24be9](https://www.etherbase.net/etherbase-token/contract-ownership)

The address above has authority over the ownable functions within the contract.

This allows the owner to call certain functions within the contract. Any compromise to the owner wallet may allow these privileges to be exploited.

We recommend:

- Establishing a Time-Lock with reasonable latency
- Assignment of privileged roles to multi-signature wallets

Contract Code Audit – Owner Accessible Functions

Function Name	Parameters	Visibility	Audit Notes
transferOwnership	address newOwner	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
burn	uint256 amount	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
increaseSupply	uint Additional_supply	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
selfDestruct	address payable ownerT	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.

The functions listed above can be called by the contract owner.

If contract ownership has been renounced there is no way for the above listed functions to be called.

Liquidity Ownership – Locked / Unlocked

No locked liquidity information has been found.



This page will contain links to locked liquidity for the project if we are able to locate that information. Locked liquidity information was not found on the project's website.

Contract Code Audit – Mint Functions

This Contract CAN Mint New NMSIS Tokens.



We do understand that sometimes mint functions are essential to the functionality of the project.

An increaseSupply function was found in the contract code.

ERC-20 Contract Code Audit – Overview

Dessert Finance was commissioned to perform an audit on

Upgradeable Proxy Nemesis Downfall (PoS) (NMSIS)

```
Submitted for verification at polygonscan.com on 2021-06-09
// File: @openzeppelin-contracts/contracts/ERC20/ERC20.sol
// SPDX-License-Identifier: MIT
pragma solidity ^0.6.0;

/**
 * @dev Provides information about the current deployment contract, including the
 * name of the transaction and its data. While these are generally available
 * via msg.sender and msg.data, they should not be exposed in public
 * functions; this module abstracts them away, exposing the account address and
 * paying for execution gas not to be the actual sender (as far as an application
 * is concerned).
 * This contract is only required for intermediate, library-like contracts.
 */
abstract contract Context {
    function msgSender() internal view virtual returns (address payable) {
        return msg.sender;
    }

    function msgData() internal view virtual returns (bytes memory) {
        // solhint-disable-next-line no-inline-assembly -- see https://github.com/OpenZeppelin/openzeppelin-contracts/pull/2000
        return msg.data;
    }
}

// File: @openzeppelin-contracts/contracts/ERC20/ERC20.sol
// SPDX-License-Identifier: MIT
pragma solidity ^0.6.0;

/**
 * @dev Interface of the ERC20 standard as defined in the EIP.
 */
interface ERC20 {
    /**
     * @dev Returns the amount of tokens in existence.
     */
    function totalSupply() external view returns (uint256);

    /**
     * @dev Returns the amount of tokens owned by 'account'.
     */
    function balanceOf(address account) external view returns (uint256);

    /**
     * @dev Moves 'amount' tokens from the caller's account to 'recipient'.
     * Returns a boolean value indicating whether the operation succeeded.
     * Emits a {Transfer} event.
     */
    function transfer(address recipient, uint256 amount) external returns (bool);

    /**
     * @dev Returns the remaining number of tokens that 'spender' will be
     * allowed to spend on behalf of 'owner' through {transferFrom}. This is
     * zero by default.
     * This value changes when {approve} or {transferFrom} are called.
     */
}
```

Contract Address

0x30a08e5E9C1C5BA21e32dFe1D66011DB3009Df8e proxy for NMSIS TOKEN
@ 0x3f5b2063e500492463589b17f9ea9cf2af9771cf

Contract Creator

0x463f64ad3448e0be80ba3b6428a9d029f25f162f

Source Code

Contract Source Code Verified

Contract Name

UChildERC20

Other Settings

default evmVersion, MIT

Compiler Version

v0.6.6+commit.6c089d02

Optimization Enabled

No with 200 runs

Code is truncated to fit the constraints of this document.

[The code in its entirety can be viewed here.](#)

The contract code is **verified** on PolygonScan.

ERC-20 Contract Code Audit – Vulnerabilities Checked

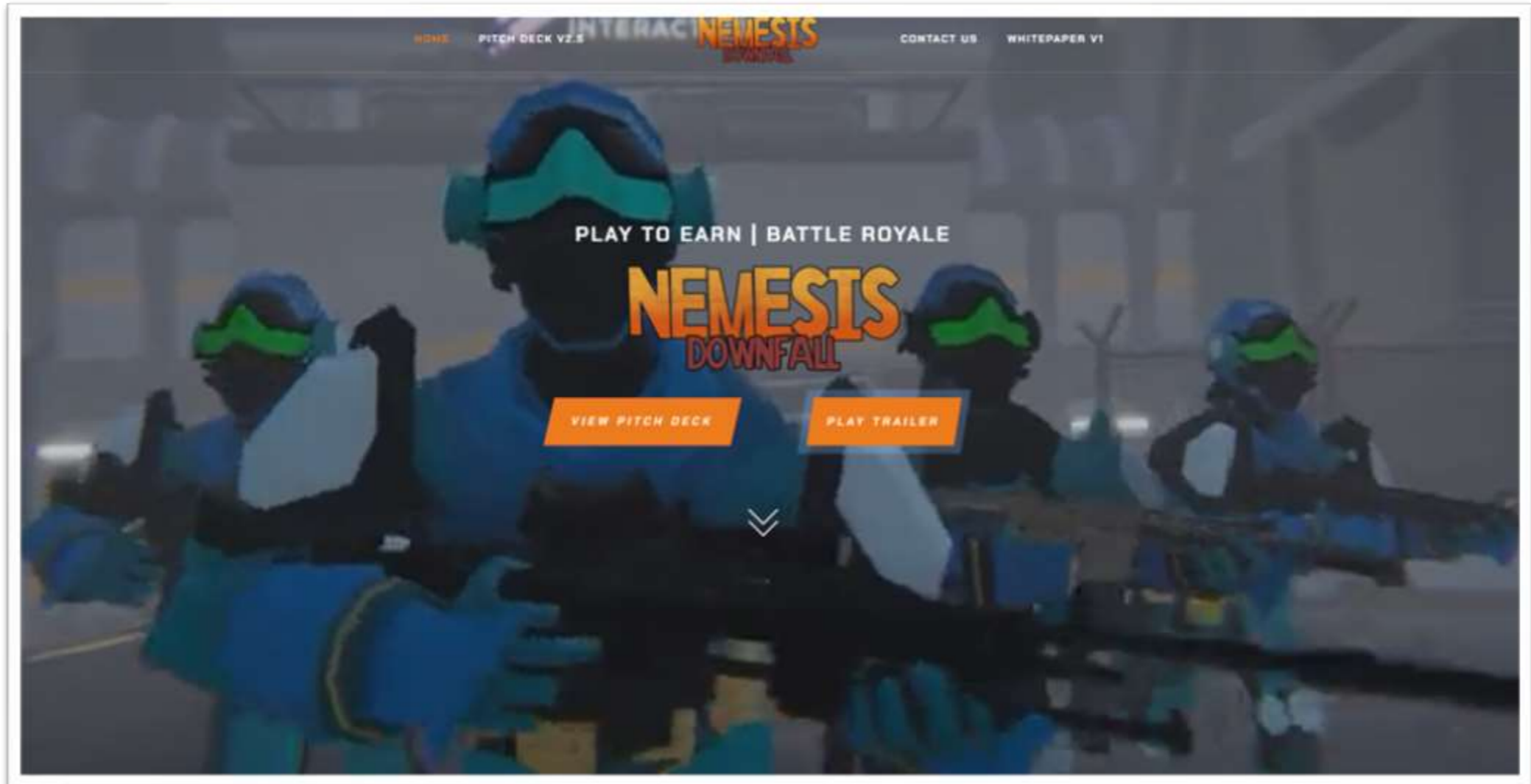
Vulnerability Tested	AI Scan	Human Review	Result
Compiler Errors	Complete	Complete	✓ Low / No Risk
Outdated Compiler Version	Complete	Complete	✓ Low / No Risk
Integer Overflow	Complete	Complete	✓ Low / No Risk
Integer Underflow	Complete	Complete	✓ Low / No Risk
Correct Token Standards Implementation	Complete	Complete	✓ Low / No Risk
Timestamp Dependency for Crucial Functions	Complete	Complete	✓ Low / No Risk
Exposed _Transfer Function	Complete	Complete	✓ Low / No Risk
Transaction-Ordering Dependency	Complete	Complete	✓ Low / No Risk
Unchecked Call Return Variable	Complete	Complete	✓ Low / No Risk
Use of Deprecated Functions	Complete	Complete	✓ Low / No Risk
Unprotected SELFDESTRUCT Instruction	Complete	Complete	✓ Low / No Risk
State Variable Default Visibility (x2)	Complete	Complete	✓ Low Risk
Deployer Can Access User Funds	Complete	Complete	✓ Low / No Risk

The contract code is **verified** on PolygonScan. CA: 0x30a08e5E9C1C5BA21e32dFe1D66011DB3009Df8e proxy for NMSIS CA 0x3f5b2063e500492463589b17f9ea9cf2af9771cf

The vulnerabilities listed above were not found in the token's Smart Contract.

Website Part 1 – Overview

www.nemesisdownfall.com



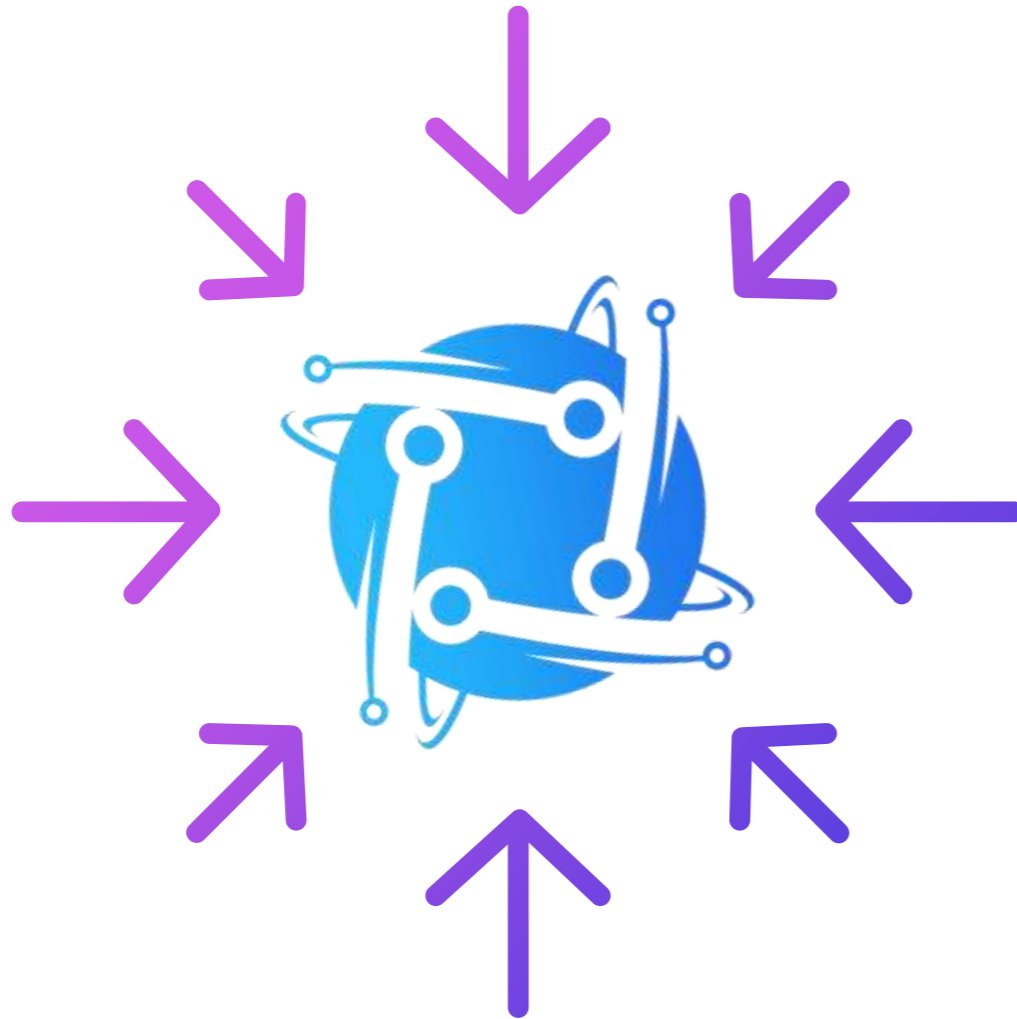
Above images are actual snapshots of the current live website of the project.

Website was registered on 10/28/2021, registration expires 10/28/2028.

✓ This **exceeds** the 3 year minimum we like to see on new projects.



Website Part 2 – Checklist



- ✓ Mobile Friendly
- ✓ No JavaScript Errors
- ✓ Spell Check
- ✓ SSL Certificate

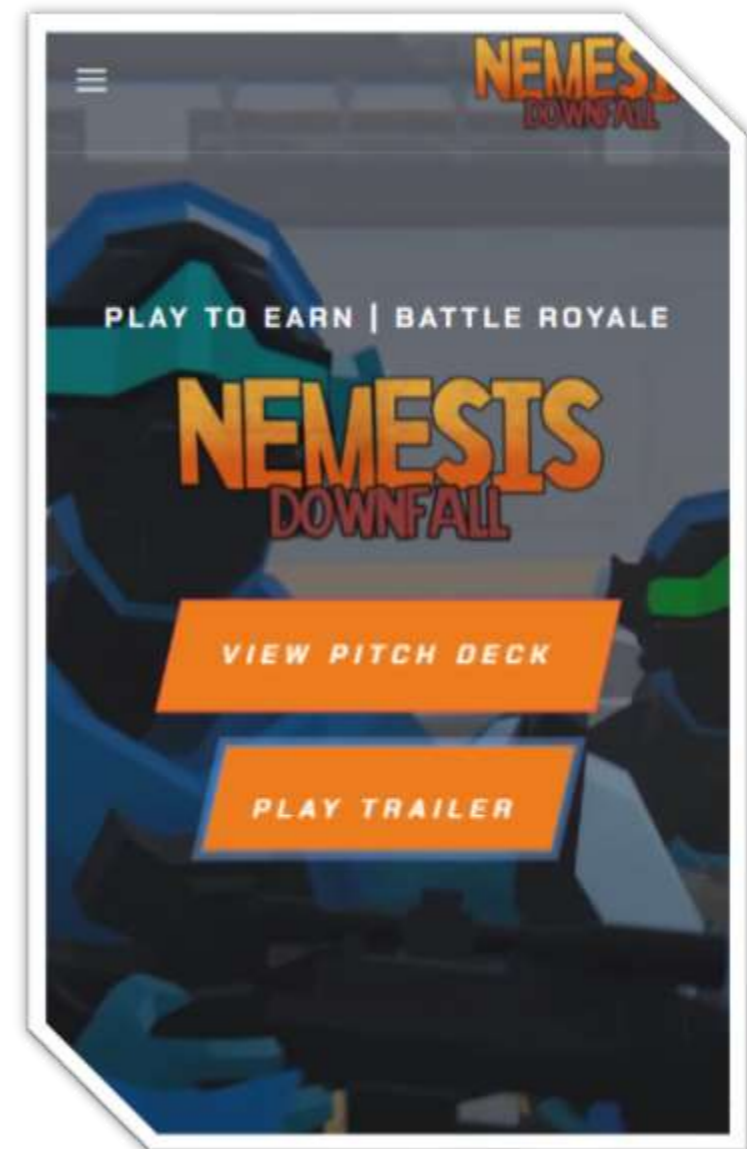
The website contained no JavaScript errors. No typos, or grammatical errors were present, and we found a valid SSL certificate allowing for access via https.

No additional issues were found on the website.

Website Part 3 – Responsive HTML5 & CSS3

No issues were found on the Mobile Friendly check for the website. All elements loaded properly and browser resize was not an issue. The team has put a considerable amount of thought and effort into making sure their website looks great on all screens.

No severe JavaScript errors were found. No issues with loading elements, code, or stylesheets.



Website Part 4 (GWS) – General Web Security



SSL CERTIFICATE

A valid SSL certificate was found. Details are as follows:

Offered to: nemesisd downfall.com

Issued by: Cloudflare Inc

Valid Until: 3/2/2023



CONTACT EMAIL

A valid contact email was found on the official website. Contact email is listed as shown below:

[Contact](mailto:info@nemesisd downfall.com)

info@nemesisd downfall.com



SPAM / MALWARE / POPUPS

No malware found

No injected spam found

No internal server errors

No popups found

Domain is marked clean by Google, McAfee, Sucuri Labs, & ESET



Social Media



We were able to locate a variety of Social Media networks for the project.

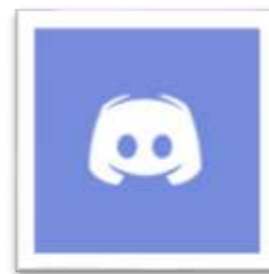
All links have been conveniently placed below.



[Twitter](#)



[Telegram](#)



[Discord](#)



[Medium](#)



[YouTube](#)

✓ **At least 3 social media networks were found.**

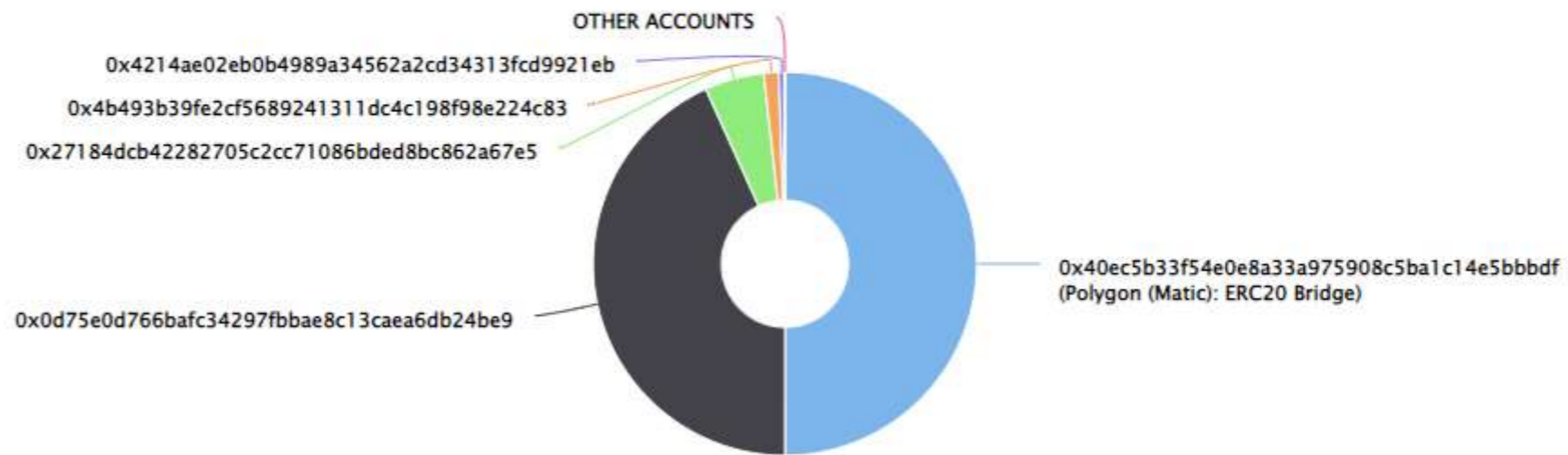
Top Token Holders

The top token holders at the time of the audit are shown below.

[Click here to view the most up-to-date list of holders](#)

Nemesis Downfall Top 100 Token Holders

Source: Etherscan.io



Rank	Address	Quantity (Token)	Percentage
1	Polygon (Matic): ERC20 Bridge	10,000,000,000	50.0000%
2	0x0d75e0d766bafc34297fbbae8c13caea6db24be9	8,650,000,000	43.2500%
3	0x27184dcb42282705c2cc71086bded8bc862a67e5	1,000,000,000	5.0000%
4	0x4b493b39fe2cf5689241311dc4c198f98e224c83	250,000,000	1.2500%
5	0x4214ae02eb0b4989a34562a2cd34313fcd9921eb	100,000,000	0.5000%

Team Overview



Cameron Geary

Co-Founder of Nemesis Downfall

Cam Geary is the founder and CEO of Dogecoin 2.0, a Binance smart chain project, which achieved a \$30,000,000 market cap only four months after its creation, and was featured on the front page of The Wall Street Journal as well as in many other notable publications including Fox Business, Yahoo Finance, NASDAQ, and The Motley Fool.

Geary has a strong background in business development, management, sales, and marketing- he is a lifelong entrepreneur and trader and a passionate economist.



Jake O'Connor

Founder of Games Interactive

Jake has been a gamer since the age of 3 and has a passion for online multiplayer games. He has a degree in computer games technology and has worked in the industry as a C++ game engine and tool developer. Jake has also worked in academia, teaching GAMES programming, and has multiple academic publications in the field. In 2020,

Jake founded Games Interactive to pursue the development of his own online games with a focus on WEB3 technology.



Charlie Char

Co-Founder of Nemesis Downfall

Charlie Char is the COO of Dogecoin 2.0. Charlie is a film producer for the last 12 years, working on small and large productions. Within his film career, he received the Northwest Regional Emmy® for Best Historical Documentary as a producer. Charlie also has started several nonprofits with the focus of supporting local filmmakers, and Children's Hospitals in Alaska. Charlie has helped raise over \$1,500,000 for kids in need in Alaska through the charity Extra Life and has managed 2 large gaming events in support of that charity. The work of supporting filmmakers and Extra Life it has awarded Charlie with the title of Top 40 under 40 in Alaska.

Business Registration

We were able to find an official registration certificate for the project. It is shown below.



Roadmap

A roadmap was found on the official website, we have conveniently placed it on this page for your viewing.



Disclaimer



The opinions expressed in this document are for general informational purposes only and are **not intended to provide specific advice or recommendations for any individual or on any specific investment**. It is only intended to provide education and public knowledge regarding projects. This audit is only applied to the type of auditing specified in this report and the scope of given in the results. Other unknown security vulnerabilities are beyond responsibility. Dessert Finance only issues this report based on the attacks or vulnerabilities that already existed or occurred before the issuance of this report. For the emergence of new attacks or vulnerabilities that exist or occur in the future, Dessert Finance lacks the capability to judge its possible impact on the security status of smart contracts, thus taking no responsibility for them. The smart contract analysis and other contents of this report are based solely on the documents and materials that the contract provider has provided to Dessert Finance or was publicly available before the issuance of this report (issuance of report recorded via block number on cover page), if the documents and materials provided by the contract provider are missing, tampered, deleted, concealed or reflected in a situation that is inconsistent with the actual situation, or if the documents and materials provided are changed after the issuance of this report, Dessert Finance assumes no responsibility for the resulting loss or adverse effects. Due to the technical limitations of any organization, this report conducted by Dessert Finance still has the possibility that the entire risk cannot be completely detected. Dessert Finance disclaims any liability for the resulting losses.

Dessert Finance provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Even projects with a low risk score have been known to pull liquidity, sell all team tokens, or exit-scam. Please exercise caution when dealing with any cryptocurrency related platforms.

The final interpretation of this statement belongs to Dessert Finance.

Dessert Finance highly advises against using cryptocurrencies as speculative investments and they should be used solely for the utility they aim to provide.



Thank You

DESSERT FINANCE PROJECT AUDIT HAS BEEN COMPLETED FOR NMSIS AT BLOCK NUMBER: **16515462**

THIS AUDIT IS ONLY VALID IF VIEWED ON [HTTPS://WWW.DSSERTSWAP.FINANCE](https://www.dessertswap.finance)

www.dessertswap.finance
<https://t.me/dessertswap>