



DESSERT
FINANCE

DogeArmy (DOGRMY)

BEP-20 Audit

Performed at block **25351401**

PERFORMED BY DESSERT FINANCE
FOR CONTRACT ADDRESS: **0xbF758F2AFEC32B92E8008b5671088D42aF616515**

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.

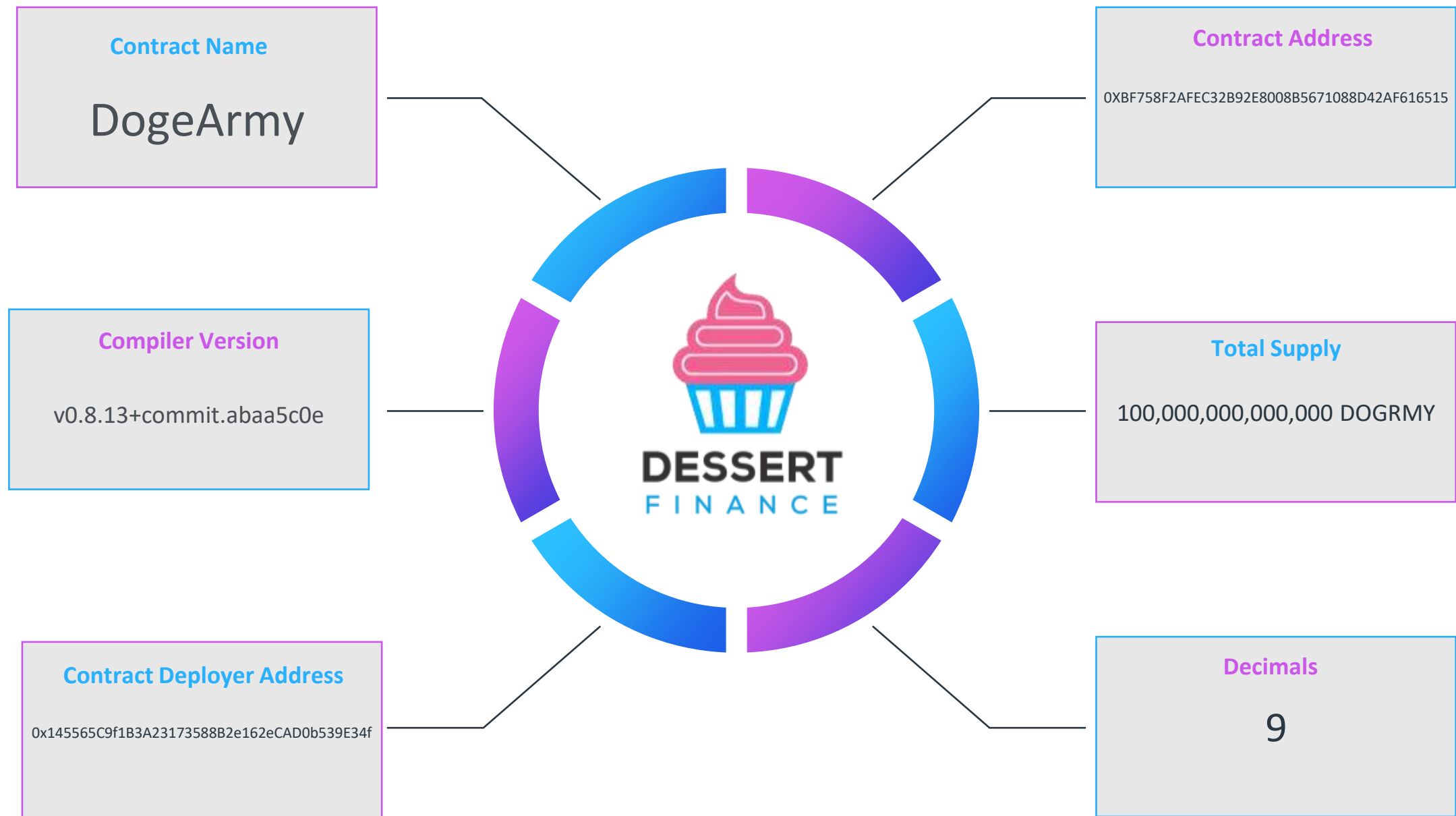
Dessert Finance in no way takes responsibility for any losses, nor does Dessert Finance encourage any speculative investments. The information provided in this audit is for information purposes only and should not be considered investment advice. Dessert Finance does not endorse, recommend, support, 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. BEP-20 Contract Code Audit – Overview
3. BEP-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. Location Audit
13. Review of Team
14. Roadmap
15. Disclaimers

Contract Code Audit – Token Overview



BEP-20 Contract Code Audit – Overview

Dessert Finance was commissioned to perform an audit on DogeArmy

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

import "@openzeppelin/contracts/access/Ownable.sol";
import "@openzeppelin/contracts/token/ERC20/ERC20.sol";
import "@openzeppelin/contracts/utils/math/SafeMath.sol";
import "@uniswap/v2-core/contracts/interfaces/IUniswapV2Factory.sol";
import "@uniswap/v2-periphery/contracts/interfaces/IUniswapV2Router02.sol";
import "../libraries/DividendTracker.sol";

contract DogeArmy is ERC20, Ownable {
    using SafeMath for uint256;

    IUniswapV2Router02 public uniswapV2Router;
    address public immutable uniswapV2Pair;

    uint8 private constant _decimals = 9;

    bool private swapping;

    DividendTracker public dividendTracker;

    address public marketingWallet;

    uint256 public maxTransactionAmount;
    uint256 public swapTokensAtAmount;
    uint256 public maxWallet;

    // sells have a higher fee
    uint256 public sellFeeIncreaseFactor = 100;

    uint256 public tradingActiveBlock = 0; // 0 means trading is not active
    mapping(address => bool) public boughtEarly; // mapping to track addresses that buy within the first 10 blocks pay a 3x tax for
    uint256 public earlyBuyPenaltyEnd; // determines when snipers/bots can sell without extra penalty

    bool public limitsInEffect = true;
    bool public tradingActive = false;
    bool public swapEnabled = false;

    // Anti-bot and anti-whale mappings and variables
    mapping(address => uint256) private _holderLastTransferTimestamp; // to hold last transfers temporarily during launch
    bool public transferDelayEnabled = true;

    address public presaleAddress;
    address public presaleRouterAddress;

    uint256 public totalFees;
    uint256 public rewardsFee;
    uint256 public marketingFee;
    uint256 public liquidityFee;

    // can be default 100 000 or to anyone who holds dividends
}
```

Contract Address

0xbF758F2AFEC32B92E8008b5671088D42aF616515

TokenTracker

DogeArmy (DOGRMY)

Contract Creator

0x145565C9f1B3A23173588B2e162eCAD0b539E34f

Source Code

Verified

Contract Name

DogeArmy

Other Settings

default evmVersion

Compiler Version

v0.8.13+commit.abaa5c0e

Optimization Enabled

Yes with 200 runs

Code is truncated to fit the constraints of this document.

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

BEP-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 BSCScan.

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:

[0x4f03019a9059ba434fd0cb18fed8edd9eda292fb](https://etherscan.io/address/0x4f03019a9059ba434fd0cb18fed8edd9eda292fb)

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
addPresaleAddressForExclusions	address _presaleAddress, address _presaleRouterAddress	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
emergencyPresaleAddressUpdate	address _presaleAddress, address _presaleRouterAddress	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
disableTransferDelay		external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
excludeFromDividends	address account	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
includeInDividends	address account	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
enableTrading		external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
updateSwapEnabled	bool enabled	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
updateMaxAmount	uint256 newNum	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
updateMaxWalletAmount	uint256 newNum	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
updateSellPenalty	uint256 selfFactor	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
excludeFromMaxTransaction	address updAds, bool isEx	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
excludeFromFees	address account, bool excluded	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
excludeMultipleAccountsFromFees	address[] calldata accounts, bool excluded	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setAutomatedMarketMakerPair	address pair, bool value	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setIsExcludedFromProcessing	address account, bool excluded	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
updateMarketingWallet	address newMarketingWallet	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
updateGasForProcessing	uint256 newValue	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
updateClaimWait	uint256 claimWait	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
removeLimits		external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
updateSwapTokensAtAmount	uint256 newAmount	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
buyBackTokens	uint256 bnbAmountInWei	external	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.

Liquidity Ownership – Locked / Unlocked

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. Please click the links for additional details.

Team.Finance

<https://www.team.finance/view-coin/0xbF758F2AFEC32B92E8008b5671088D42aF616515?name=DogeArmy&symbol=DOGRMY>

Contract Code Audit – Mint Functions

This Contract Cannot Mint New DOGRMY Tokens.



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

A mint function was not found in the contract code.

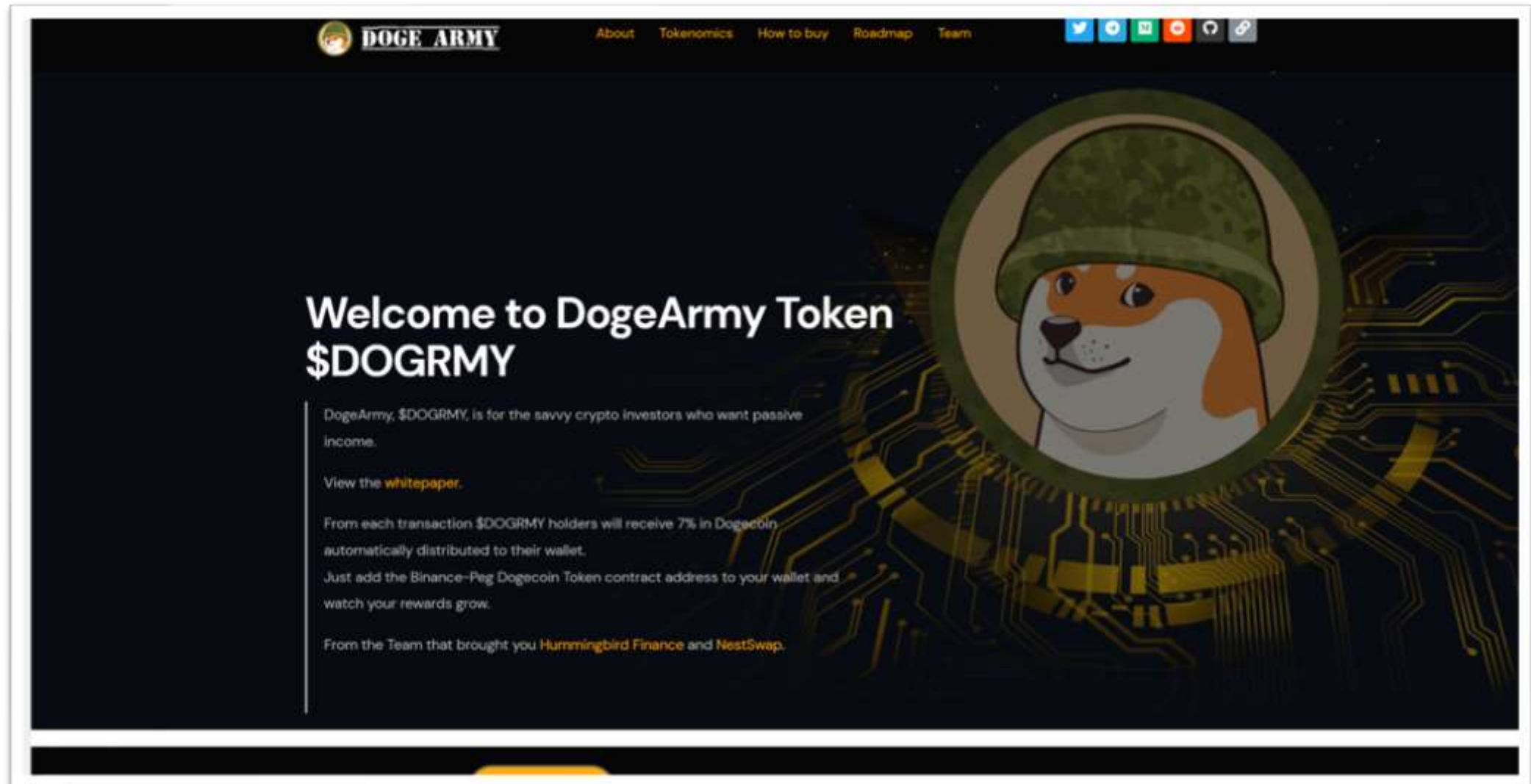
Contract Transaction Fees

At the time of Audit the transaction fees (“tax”) listed below are the fees associated with trading. These fees are taken from every buy and sell transaction unless otherwise stated.



Website Part 1 – Overview

www.dogearmy.app



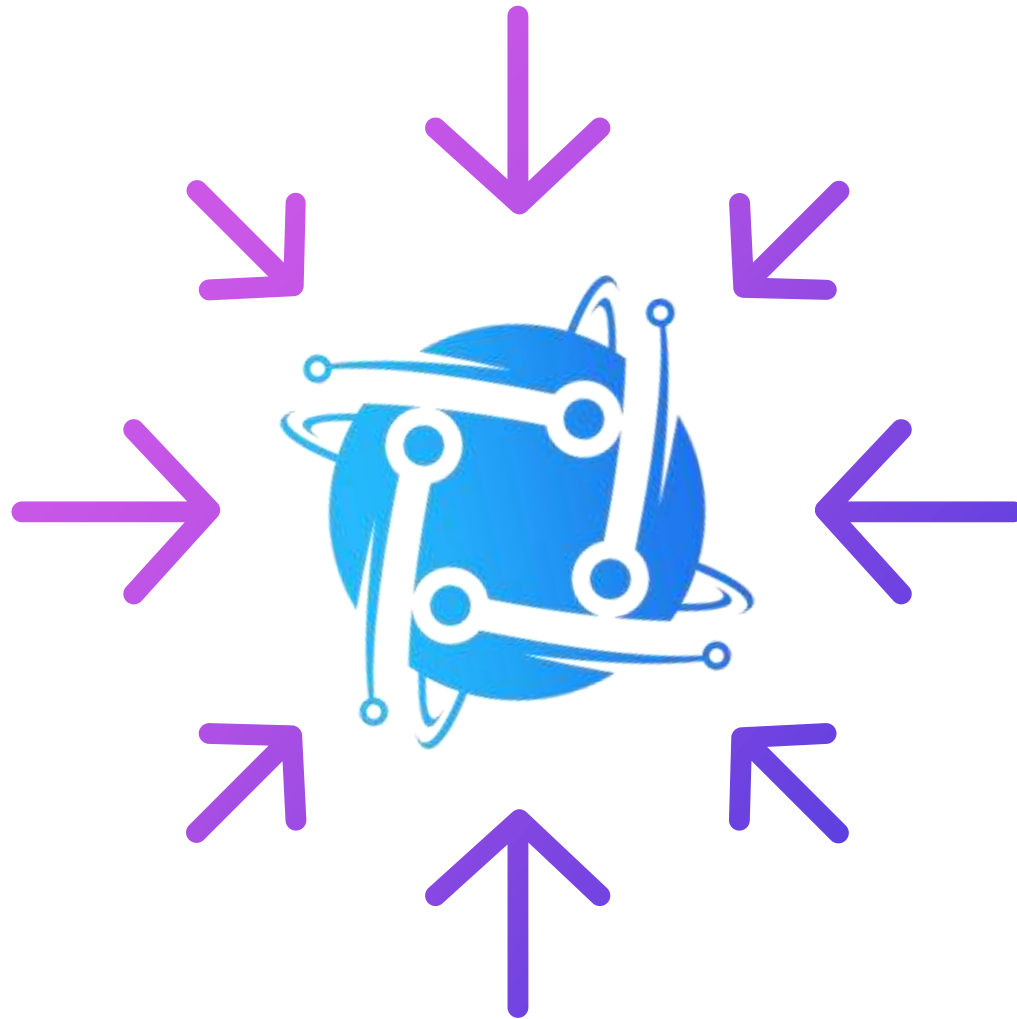
Above images are actual snapshots of the current live website of the project.

Website was registered on 05/13/2022, registration expires 05/13/2025.

✓ This meets 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 major 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. Minor JavaScript errors were found.

```
Uncaught ReferenceError: jQuery is not defined
    at core.min.js?ver=1.13.2:5:88
    at core.min.js?ver=1.13.2:5:96

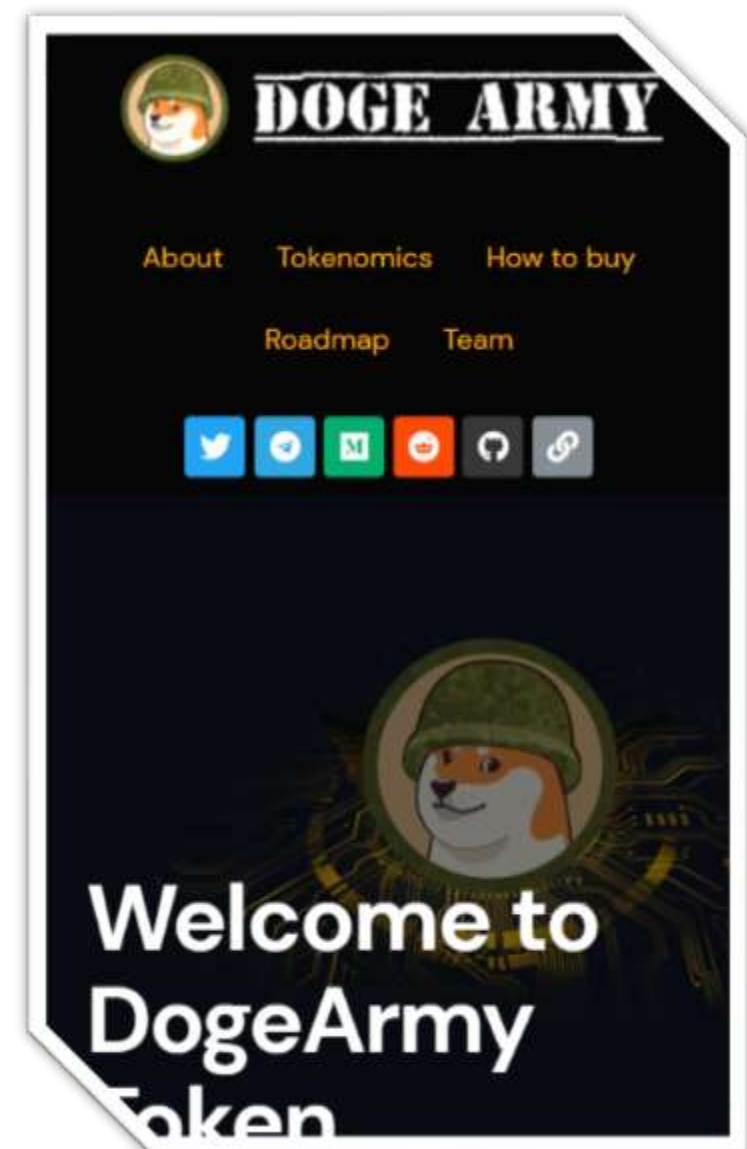
Uncaught ReferenceError: ep-sound-effects.min.js?ver=6.0.11:1
    at ep-sound-effects.min.js?ver=6.0.11:1:2148

Uncaught ReferenceError: jQuery is not defined
    at ep-advanced-icon-box.min.js?ver=6.0.11:1:426

Uncaught ReferenceError: jQuery is not defined
    at ep-accordion.min.js?ver=6.0.11:1:1346

Uncaught ReferenceError: jQuery is not defined
    at ep-timeline.min.js?ver=6.0.11:1:304

Uncaught ReferenceError: frontend.min.js?ver=3.10.3:2
    at 171 (frontend.min.js?ver=3.10.3:2:9484)
    at __webpack_require__ (webpack-pro.runtime.js?ver=3.10.3:1:17)
```



Website Part 4 (GWS) – General Web Security



SSL CERTIFICATE

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

Offered to: dogearmy.app

Issued by: R3

Valid Until: 04/08/2023



CONTACT EMAIL

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

[Contact](mailto:support@dogearmy.app)

support@dogearmy.app



SPAM / MALWARE / POPUPS

No malware found

No injected spam found

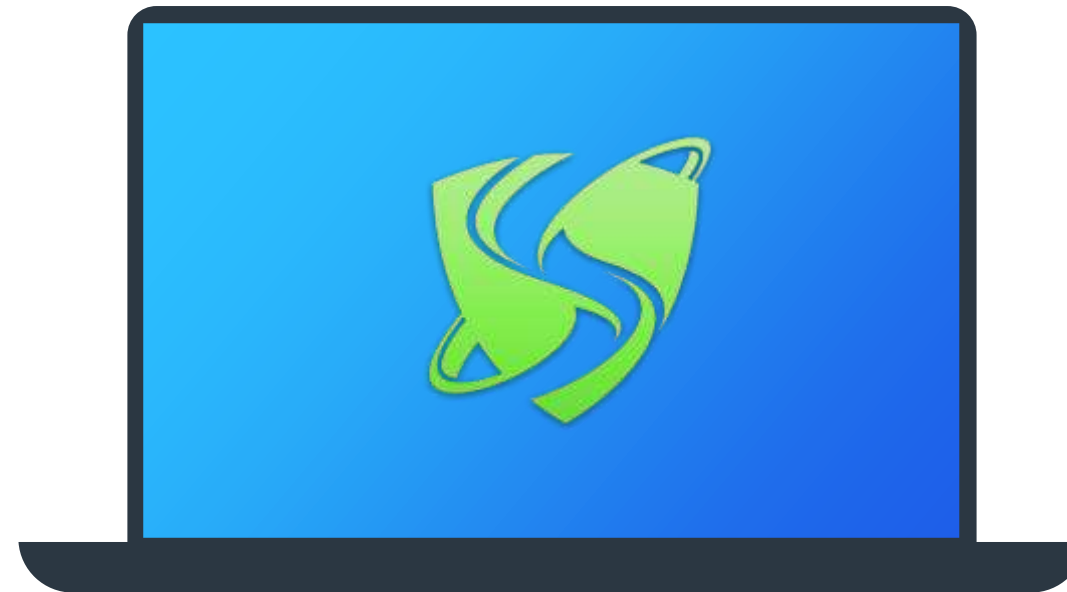
No internal server errors

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](#)



[Medium](#)



[Reddit](#)

✓ At least 3 social media networks were found.

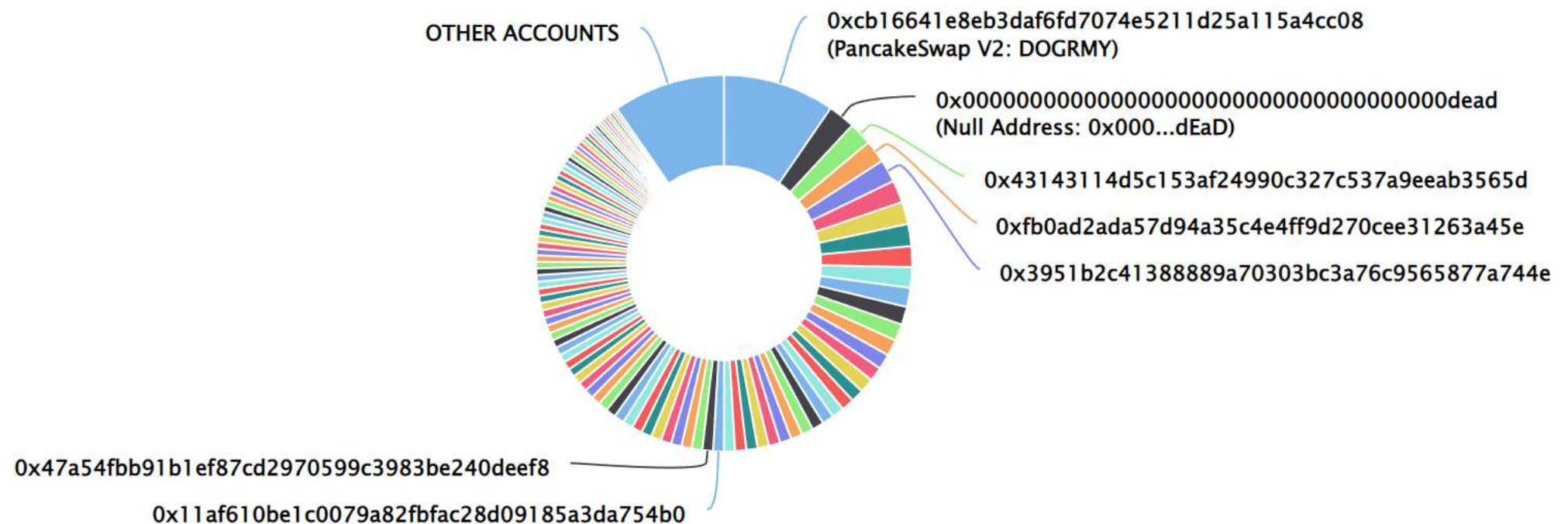
Top Token Holders

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](#)

DogeArmy Top 100 Token Holders

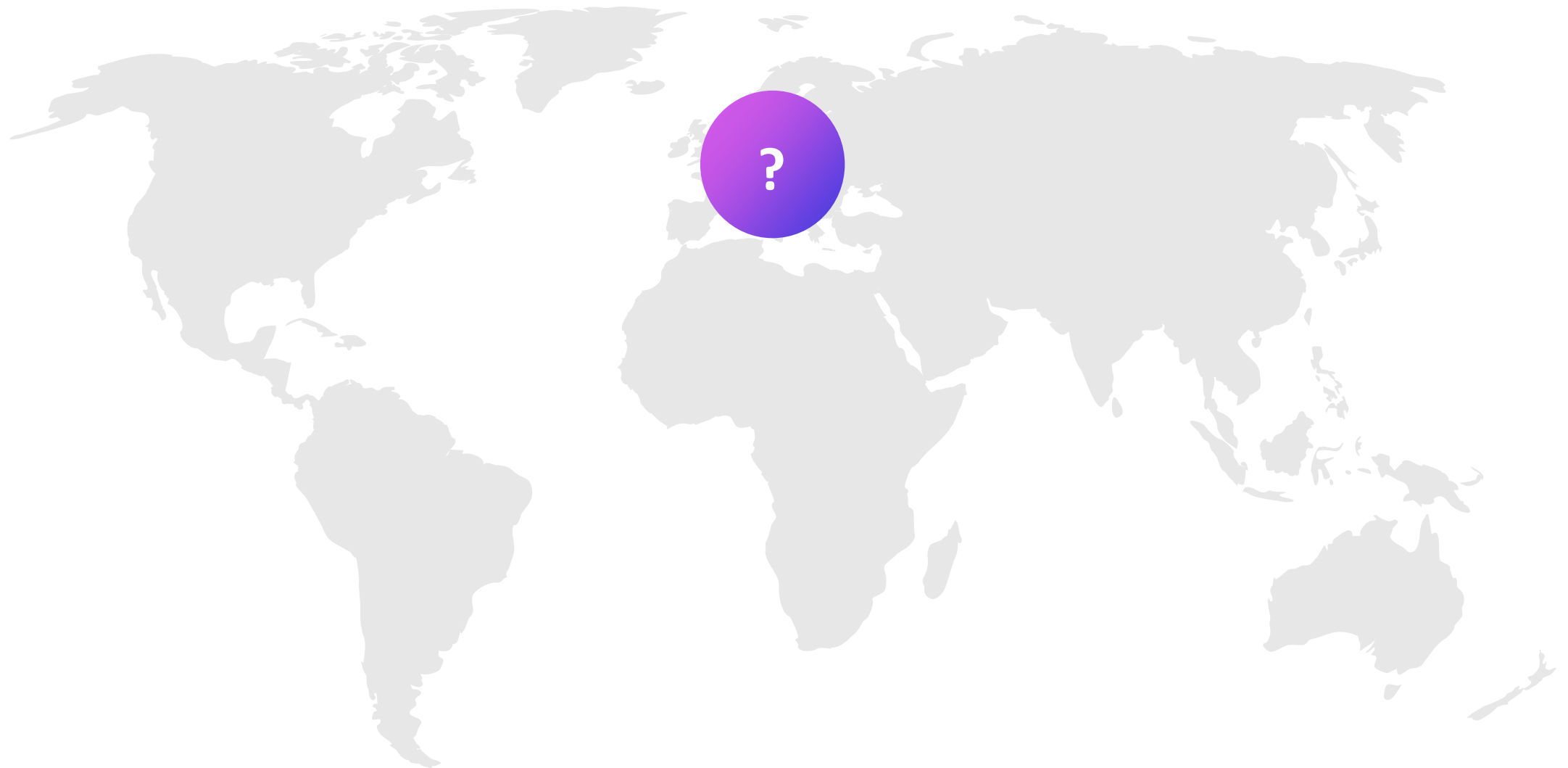
Source: BscScan.com



Rank	Address	Quantity (Token)	Percentage
1	PancakeSwap V2: DOGRMY	9,592,876,105,414.020947113	9.5929%
2	Null Address: 0x000...dEaD	2,344,451,182,677.775188227	2.3445%
3	0x43143114d5c153af24990c327c537a9eeab3565d	1,978,821,450,444.093988888	1.9788%
4	0xfb0ad2ada57d94a35c4e4ff9d270cee31263a45e	1,953,915,419,202.332243708	1.9539%
5	0x3951b2c41388889a70303bc3a76c9565877a744e	1,953,290,305,786.576849169	1.9533%


Location Audit

We were unable to identify a primary location for the project at this time or a location has not been declared.




Team Overview


The following information about the team was found on the projects website.



Gideon
GENERAL


The founder of DogeArmy and Hummingbird Finance. In charge of strategic development and management and operations of DogeArmy project.






Bryden
MAJOR


CEO at Hummingbird Finance. Planning tactical operations, support and personnel.





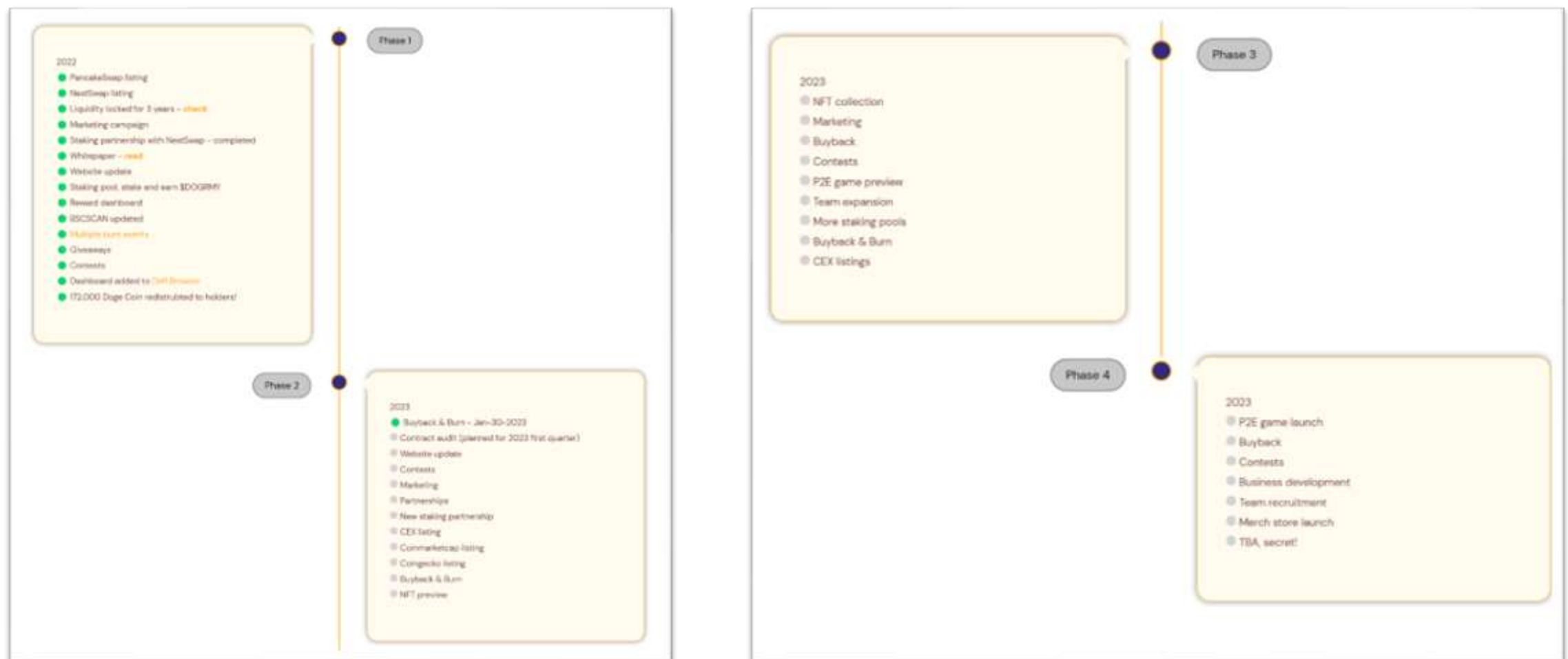
Finn
LUETENANT

Chief financial officer.



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 DOGEARMY (DOGRMY) AT BLOCK NUMBER: **25351401**

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

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