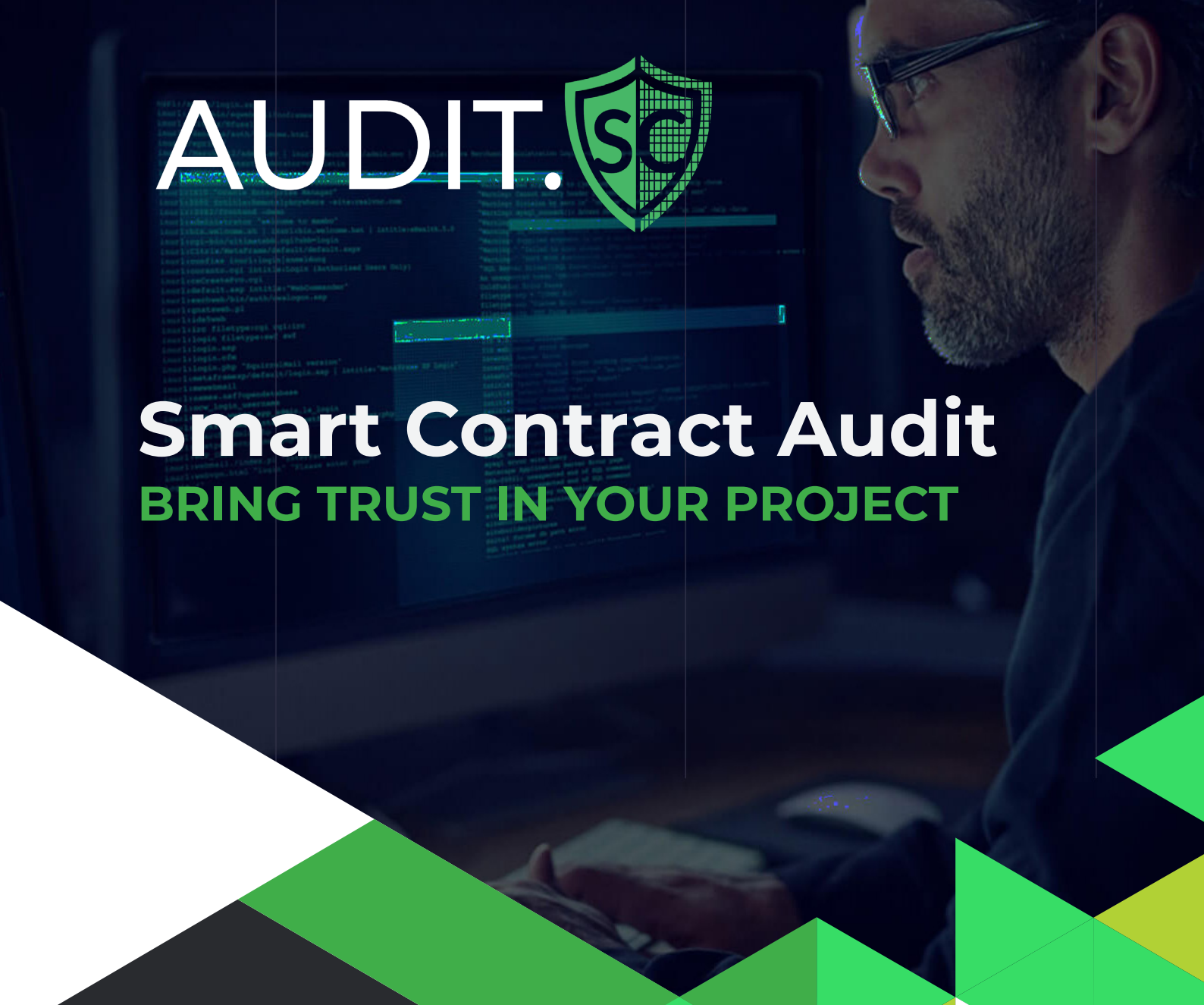

AUDIT.SC

Smart Contract Audit

BRING TRUST IN YOUR PROJECT

AUDIT-SC PARTNER DOGE PUNKS

WWW.AUDIT.SC

2021

DogePunks

FULL SMART CONTRACT AUDIT

SOLIDITY CHECK

Audit SC Guarantees that every smart contract that has been audited has gone through both automated Smart Contract Scanner Softwares and is manually verified by one of our highly experienced smart contract experts.

Table of Contents

AUDIT-SC

DISCLAIMER

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the below disclaimer below – please make sure to read it in full.

DISCLAIMER: By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and AUDIT-SC and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (AUDIT-SC) owe no duty of care towards you or any other person, nor does AUDIT-SC make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided «as is», without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and AUDIT-SC hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, AUDIT-SC hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against AUDIT-SC, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report.

OVERVIEW

PROJECT SUMMARY

Project Name	**DogePunks**
Platform	Binance Smart Chain
Language	Solidity

AUDIT SUMMARY

Date	01-11-2021
Audit Type	Static Analysis, Manual Review
Audit Result	**FAILED - NOT SAFE**

RISK SUMMARY

Risk Level	Total	Found	Pending	Solved	Acknowledgde	Objected
Critical	1	1	0	0	1	0
Major	1	1	1	0	0	0
Medium	0	0	0	0	0	0
Minor	0	0	0	0	0	0
Informative	2	2	2	0	0	0
Discussion	0	0	0	0	0	0

FINDINGS

Function Default Visibility

SWC-ID: SWC-100

Relationship:
CWE-710: Improper Adherence to Coding Standards

Description:
Functions that do not have a function visibility type specified are public by default. This can lead to a vulnerability if a developer forgot to set the visibility and a malicious user is able to make unauthorized or unintended state changes or unnecessary gas usage.

Relevance:
public functions that are never called by the contract should be declared external to save gas.

Category	Risk Level	Number of Findings	Status
SWC-100	Informative	**2**	Pending

Business Logic Error

Description:
The desired business logic can not be enforced, or can be ignored

Relevance:

DogePunk NFT's secondary sales do not yield any rewards to the rewardspool

Category	Risk Level	Number of Findings	Status
Coding Error	Critical	**1**	Unmitigated

Uncommented Code

Description:
Large pieces of code without explanatory comments can be difficult to understand

Relevance:

The majority of the contract nft_contract.sol has no comments, making it difficult for humans to understand what the code is trying to do

Category	Risk Level	Number of Findings	Status
Transparency	Major	**1**	Pending

AUDIT DETAILS

SCW-100 Function Default Visibility

contractURI() should be declared external

name() should be declared external:

Uncommented Code

Below is a snapshot of the main contract, nft_contract.sol, with next to no relevant comments in 443 lines of code

```
342   function claimRewardAuto(uint256 tokenId) internal returns (uint256) {
343     uint256 balance = getReflectionBalance(tokenId);
344     uint256 lastDividend = lastDividendAt[tokenId];
345
346     if (balance > 0) {
347       totalDividendDistributed += balance;
348       lastDividendAt[tokenId] = totalDividend;
349       userReward[ownerOf(tokenId)].totalAmount += balance;
350       userReward[ownerOf(tokenId)].lastAmount = balance;
351       userReward[ownerOf(tokenId)].lastTime = block.timestamp;
352
353       if (hasCustomClaimToken(tokenId) && autoRewardAsCustomIsActive) {
354         bool success = swapEthForCustomToken(ownerOf(tokenId), tokenId, balance);
355         if (!success) {
356           (bool secondSuccess, ) = ownerOf(tokenId).call{value: balance, gas: 3000}('');
357           if (!secondSuccess) {
358             lastDividendAt[tokenId] = lastDividend;
359             totalDividendDistributed -= balance;
360             userReward[ownerOf(tokenId)].totalAmount -= balance;
361             userReward[ownerOf(tokenId)].lastAmount = 0;
362             return 0;
363           }
364         }
365       } else {
366         payable(ownerOf(tokenId)).transfer(balance);
367       }
368     }
369     return balance;
370   }
371
372   function swapEthForCustomToken(
373     address user,
374     uint256 id,
375     uint256 amt
376   ) internal returns (bool) {
377     address[] memory path = new address[](2);
378     path[0] = uniswapV2Router.WETH();
379     path[1] = nftCustomClaimToken[id];
380     try uniswapV2Router.swapExactETHForTokens{value: amt}(0, path, user, block.timestamp + blockDelayCustomReward) {
381       return true;
382     } catch {
383       return false;
384     }
385   }
386
387   function currentRate() public view returns (uint256) {
388     if (totalSupply() == 0) return 0;
```

Business Logic Error

Our initial findings concluded a critical vulnerability in the enforcement of the marketed business logic by the DogePunks Team. During our manual review, we reported to the excessively anonymous developer, in writing, our concerns about the enforcement of the fundamental benefit that DogePunk proposes, namely the sales-fee distribution of secondary sales from any DogePunk NFT sale. Our submission to their team was as follows:

Critical business logic vulnerability found.:
Improper enforcement of business logic.

The DogePunk NFTs intend a reward of a salesFee to all NFT holders at the point of any sale, throughout the lifecycle of DogePunks. This is initially enforced through their mint() function at the time of offering. When a token is sold, it is minted directly to the buyer's wallet, and the funds are distributed to the current holders of the DogePunks.

The issue arises after the initial sale has been concluded, and the minted token is effectively in the wallet of the buyer.

Scenario:

Owner of DogePunk with ID #999 has successfully purchased the DogePunk through the initial offer for the amount of 10BNB, of which 12% is successfully distributed to all current tokenholders. Now, an interested buyer offers Owner of #999 11BNB for the NFT.

Due to the salesfee on this transaction through a marketplace that adheres to the proposed method of transfer for DogePunks, this would result in a net loss for Owner of #999, since a certain % is subtracted from the purchase amount.

Exploit

If marketplace www.avoid-nft-tax.io offers up a solution as a middleman, allowing the seller to fund a contract for 11BNB, and Owner of #999 to transfer DogePunk #999 to a swap contract, the transfer function of the NFT would be used and executed successfully. This would result in Owfer of #999 to sell the NFT for 11BNB, and the buyer to receive the DogePunk #999 while avoiding any sales tax.

Current Status: Unmitigated

After several meetings with the core team and a high-stakes holder of DogePunks, we explained in detail that one of the key benefits for DogePunk NFT's is the re-distribution of a percentage of ANY sale, not just the initial mint, and that with our given scenario a substantial benefit for current investors would fall through due to a simple coding error.

When the DogePunks team agreed that this was indeed a critical oversight, they promised to make efforts to mitigate the risks and fix the vulnerability.

Unfortunately, they chose a deceptive route with their following announcement:

DogePunks NFT Contract Update Communication

Dear DogePunks NFT Community,

Thank you for your patience with us as we navigated recent questions regarding market place fees and their allocation to the DogePunks NFT rewards pool.

Our team consulted with an external dev team (the Defi Sepculate group) and we updated our contract to route to the DogePunks NFT contract ID vs a deployer wallet:

https://api.doge-punks.io/api/contract

When DogePunks NFTs are listed on external marketplaces, the marketplace will process the transaction based on the above contract details.

All current marketplaces that list DogePunks NFTs are ERC-721 compliant. ERC-721 is a standard of regulatory code that all major NFT market places abide by.

The ERC-721 (Ethereum Request for Comments 721), proposed by William Entriken, Dieter Shirley, Jacob Evans, Nastassia Sachs in January 2018, is a Non-Fungible Token Standard that implements an API for tokens within Smart Contracts.

Currently, DogePunks NFTs are listed with lootex.io and NFTrade.com (both of which are ERC-721 compliant) and will process sales transactions by routing funds to our contract address, which will then be deposited into the rewards pool.

A “hypothetical” 0% marketplace does not exist and would not be compliant with ERC-721 standards. If such a marketplace was created, it would be in violation of ERC-721 regulations, and no NFT project would list with a marketplace in violation of these standards.

Thank you for your patience with us as we navigated these questions regarding secondary marketplace transactions.

BUT THE GOOD NEWS IS THIS: DogePunks NFT secondary marketplace sales have always routed back to the contract and will continue to do so

Miggy, Cryptik, Bushey, and the rest of the DogePunks NFT team.

Much to the surprise and distaste of our security analyst, the published announcement contains arguably deliberately deceptive and false information. The majority of given statements are unfounded, partially true, or indisputably false.

Though they hold no obligation to consult with their initial security auditor, the announcement that was made without having it checked by the original finder of the bug raises questions. Particularly about the intent and ethics behind the released statement.

Our Findings:

In order to understand the nature of the announced solution, we test it against the original vulnerability. The scenario as reported to DogePunks is, according to their statement, mitigated. Their supportive context of this is summed up in the three following statements:

- We updated our contract to route to the DogePunks NFT contract ID vs a deployer wallet - All current marketplaces that list DogePunks NFTs are ERC-721 compliant. ERC-721 is a standard of regulatory code that all major NFT market places abide by.

- A “hypothetical” 0% marketplace does not exist and would not be compliant with ERC-721 standards. If such a marketplace was created, it would be in violation of ERC-721 regulations, and no NFT project would list with a marketplace in violation of these standards.

Below, we investigate the supportive statements on their merits:

Statement:

“

we updated our contract to route to the DogePunks NFT contract ID vs a deployer wallet:

”

Our refutation

The above statement is very ambiguous and does not explain at all why or how it mitigates the financial risk that arises from the reported vulnerability. The code snipped in their API holds absolutely no bearing as to the validity of the claim that the vulnerability is fixed.

This solution would add a substantial centralization risk to the project, since the API and the supposed fix it points to, can be changed at any given moment. In this case, anyone with access to the server where the API resides, could swap out the information for their own financial gain at the expense of DogePunk Holders.

Statement:

“

All current marketplaces that list DogePunks NFTs are ERC-721 compliant. ERC-721 is a standard of regulatory code that all major NFT marketplaces abide by.

“

Our refutation:

This statement itself is constructed in its entirety by false statements. There is no ERC-721 compliance for marketplaces, nor is there a governing body to enforce this compliance- even less so in regards to the context of DogePunks sales fee. It most certainly is not a piece of regulatory code for marketplaces to abide by. It is a standardized interface for Non-Fungible Tokens (NFTs).

Statement:

“

A “hypothetical” 0% marketplace does not exist and would not be compliant with ERC-721 standards. If such a marketplace was created, it would be in violation of ERC-721 regulations, and no NFT project would list with a marketplace in violation of these standards.

”

Our refutation:

This statement in itself shows the deliberate deception DogePunks has employed towards its community. By claiming that there is an overarching regulatory instance that protects against the financial loss that holders would incur, they have engaged in what we can only assume is an attempt to misguide prospective buyers into investing in their project.

The claim that the hypothetical nature of the exploit invalidates is, offers only more evidence as to the recklessness DogePunk’s core team and developer are showing. The precise point of a security audit is to find, prevent, mitigate, or solve hypothetical issues before they become real, at which point all NFT holders would be left with potentially immense financial losses.

Conclusion

As of the moment the aforementioned publication was made by the DogePunks team, we deem their claims beyond uneducated: they are flat-out lies.

The unethical circumstances in which the events leading up to their announcements transpired leave no room for any conclusion but to say that the misguiding and deceptive statements are done with the full knowledge of them being so.

We condemn misleading investors in the strongest of terms and take strong exception to ethics that the core team of DogePunks has been lacking.

For failing to uphold their own claims and deceiving their community, we strongly advise against investing in DogePunks at the time of this writing.

AUDIT.SC

CONTACT US

Audit SC Guarantees that every smart contract that has been audited has gone through both automated Smart Contract Scanner Softwares and is manually verified by

E-mail: info@audit.sc

Website: www.audit.sc

AUDIT.SC