Cyberscope

Audit Report

ShieldDAO Token

August 2022

Type	BEP20
Network	BSC
Address	0x0765797650e2d13EEaC723Bc31a11929EB1a664b
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	SDD
Compiler Version	v0.8.7+commit.e28d00a7
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0x0765797650e2d13EEaC723Bc31a11929EB1a664b
Symbol	SDD
Decimals	9
Total Supply	300,000
Domain	https://www.shielddao.financial

Source Files

Filename	**SHA256**
contract.sol	dbe5e98cc115a3bba818ebfc07a2c1d84506faadaa3984d66781a35fa239ba8f

Audit Updates

Initial Audit	18th July 2022
Corrected Phase 1	23rd July 2022
Corrected Phase 2	20th August 2022

Contract Analysis

🔴 Critical 🟠 Medium ⚪ Minor 🔵 Pass

Severity	Code	Description	Status
🔵	ST	Stops Transactions	Passed
🔵	OCTD	Transfers Contract's Tokens	Passed
⚪	OTUT	Transfers User's Tokens	Unresolved
🔵	ELFM	Exceeds Fees Limit	Passed
⚪	ULTW	Transfers Liquidity to Team Wallet	Unresolved
🔵	MT	Mints Tokens	Passed
🔵	BT	Burns Tokens	Passed
🔵	BC	Blacklists Addresses	Passed

OTUT - Contract Owner has the authority to transfer the user's tokens

Criticality	minor
Location	contract.sol#L332
Status	Unresolved

Description

The contract owner has the authority to transfer the balance of a user’s contract to the owner’s contract. The owner may take advantage of it by calling the `claimToken` function.

```
function claimToken(address token, uint256 amount) public {
       IERC20(token).transfer(FundAddress, amount);
}
```

Recommendation

The team should carefully manage the private keys of the owner’s account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. That risk can be prevented by temporarily locking the contract or renouncing ownership.

ULTW - Contract Owner has the authority to increase the amount of liquidity taken by dev wallet more than a reasonable percent

Criticality	minor
Location	contract.sol#L328
Status	Unresolved

Description

The contract owner has the authority to transfer funds without limit to the team wallet. These funds have been accumulated from fees collected from the contract. The owner may take advantage of it by calling the `claimBalance` .

```
function claimBalance() public {
        payable(FundAddress).transfer(address(this).balance);
}
```

Recommendation

The contract could embody a check for the maximum amount of funds that can be swapped. Since a huge amount may volatile the token's price.

The team should carefully manage the private keys of the owner's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. That risk can be prevented by temporarily locking the contract or renouncing ownership.

Contract Diagnostics

● Critical ● Medium ● Minor

Severity	Code	Description	Status
●	STC	Succeeded Transfer Check	Unresolved
●	L01	Public Function could be Declared External	Unresolved
●	L02	State Variables could be Declared Constant	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L14	Uninitialized Variables in Local Scope	Unresolved

STC - Succeeded Transfer Check

Criticality	minor
Location	contract.sol#L333
Status	Unresolved

Description

According to the ERC20 specification, the transfer methods should be checked if the result is successful. Otherwise, the contract may wrongly assume that the transfer has been established.

```
IERC20(token).transfer(FundAddress, amount);
```

Recommendation

The contract should check if the result of the transfer methods is successful.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L219,324,99,228,237,90,328,104,206,210
Status	Unresolved

Description

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

```
allowance
claimBalance
renounceOwnership
approve
transferFrom
owner
claimToken
transferOwnership
balanceOf
...
```

Recommendation

Use the external attribute for functions never called from the contract.

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L122
Status	Unresolved

Description

Constant state variables should be declared constant to save gas.

```
fee
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L125,131,124
Status	Unresolved

Description

Solidity defines a naming convention that should be followed. Rule exceptions:

- Allow constant variable name/symbol/decimals to be lowercase.
- Allow _ at the beginning of the mixed_case match for private variables and unused parameters.

```
FundAddress
_swapRouter
FeeAddress
```

Recommendation

Follow the Solidity naming convention.
https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions.

L14 - Uninitialized Variables in Local Scope

Criticality	minor
Location	contract.sol#L291
Status	Unresolved

Description

The are variables that are defined in the local scope and are not initialized.

```
feeAmount
```

Recommendation

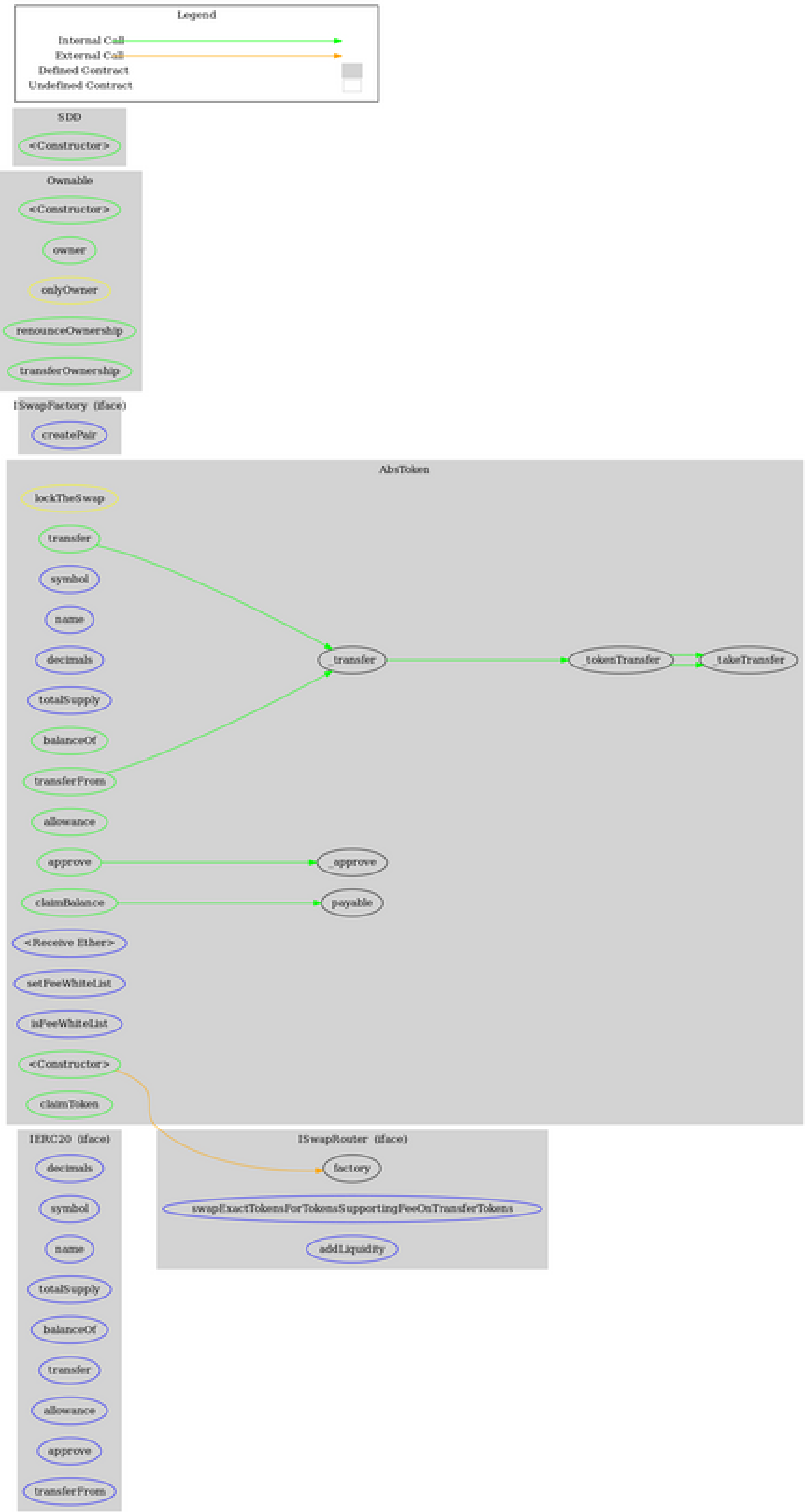
All the local scoped variables should be initialized.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
IERC20	Interface			
	decimals	External		-
	symbol	External		-
	name	External		-
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
ISwapRouter	Interface			
	factory	External		-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	addLiquidity	External	✓	-
ISwapFactory	Interface			
	createPair	External	✓	-
Ownable	Implementation			
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
AbsToken	Implementation	IERC20, Ownable		
	<Constructor>	Public	✓	-

	symbol	External		-
	name	External		-
	decimals	External		-
	totalSupply	External		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	_approve	Private	✓	
	_transfer	Private	✓	
	_tokenTransfer	Private	✓	
	_takeTransfer	Private	✓	
	<Receive Ether>	External	Payable	-
	setFeeWhiteList	External	✓	onlyOwner
	isFeeWhiteList	External		-
	claimBalance	Public	✓	-
	claimToken	Public	✓	-
SDD	Implementation	AbsToken		
	<Constructor>	Public	✓	AbsToken

Contract Flow



Domain

Domain Name	shielddao.financial
Registry Domain ID	515e85c778ba473595d63f205b5456f9-DONUTS
Creation Date	2022-06-28T08:00:31Z
Updated Date	2022-07-03T08:01:03Z
Registry Expiry Date	2023-06-28T08:00:31Z
Registrar WHOIS Server	whois.godaddy.com/
Registrar URL	http://www.godaddy.com/domains/search.aspx?ci=8990
Registrar	GoDaddy.com, LLC
Registrar IANA ID	146

The domain was created 11 months before the creation of the audit.

There is no public billing information, the creator is protected by the privacy settings.

Summary

There are some functions that can be abused by the owner like transferring the user's tokens and transferring funds to the team’s wallet. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats. The fees are fixed to 3%.

Disclaimer

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment.

Cyberscope team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed.

The Cyberscope team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Cyberscope receive a payment to manipulate those results or change the awarding badge that we will be adding in our website.

Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token.

The Cyberscope team disclaims any liability for the resulting losses.

About Cyberscope

Coinscope audit and K.Y.C. service has been rebranded to Cyberscope.

Coinscope is the leading early coin listing, voting and auditing authority firm. The audit process is analyzing and monitoring many aspects of the project. That way, it gives the community a good sense of security using an informative report and a generic score.

Cyberscope and Coinscope are aiming to make crypto discoverable and efficient globally. They provide all the essential tools to assist users draw their own conclusions.

The Cyberscope team

https://www.cyberscope.io