COINSCOPE

Audit Report

AltSwitch

January 2022

Type	BEP20
Network	BSC
Address	0x2ec79904C2aB4F8b6e8e89c743CB7F7a88DFc0fE
Audited by	© coinscope

Table of Contents

Contract Review

Contract Name	AltSwitch
Compiler Version	v0.8.10+commit.fc410830
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0x2ec79904C2aB4F8b6e8e89c743CB7F7a88DFc0fE
Symbol	ALTS
Decimals	9
Total Supply	1,000,000,000
Source	contract.sol
Domain	altswitch.io

Audit Updates

Initial Audit	14th January 2022
Corrected	18 March 2022

Contract Analysis

🔴 Critical 🟠 Medium ⚪ Minor 🔵 Pass

Severity	Code	Description	Status
🔴	ST	Contract Owner is not able to stop or pause transactions	Acknowledged
🔵	OCTD	Contract Owner is not able to transfer tokens from specific address	
🔵	OTUT	Owner Transfer User's Tokens	
🔴	ELFM	Contract Owner is not able to increase fees more than a reasonable percent (25%)	Acknowledged
🔵	ULTW	Contract Owner is not able to increase the amount of liquidity taken by dev wallet more than a reasonable percent	
🔵	MT	Contract Owner is not able to mint new tokens	
🔵	BT	Contract Owner is not able to burn tokens from specific wallet	
🟠	BC	Contract Owner is not able to blacklist wallets from selling	Acknowledged

ST - Stop Transactions

Criticality	critical
Location	contract.sol#L1470

Description

The contract owner has the authority to stop the buy operations for all users excluding the owner. The owner may take advantage of it by setting the `maxSellPercentage` to zero.

```
if(!_isExcludedFromFees[from] && !_isExcludedFromFees[to] &&
!automatedMarketMakerPairs[from]){
    checkMaxSell(amount);
...

// checkMaxSell()
require(amount <= (liqSupply * maxSellPercentage) / 100, "You are exceeding
maxSellPercentage");
```

Recommendation

The contract could embody a check for not allowing setting the _maxTxAmount less than a reasonable amount. A suggested implementation could check that the maximum amount should be more than a fixed percentage of the total supply.

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.

Team Update

The team has acknowledged the threat and transferred the contract ownership to a multi-sign mechanism.

ELFM - Exceed Limit Fees Manipulation

Criticality	critical
Location	contract.sol#L1239,L1246

Description

The contract owner has the authority to increase over the allowed limit of 25%. The owner may take advantage of it by calling the `updateSellFees` or the `updateFees` function with a high percentage value.

```
function updateSellFees(uint256 _rewards, uint256 _liquidity, uint256
_operation) external onlyOwner {
    sellRewardsFee = _rewards;
    sellLiquidityFee = _liquidity;
    sellOperationFee = _operation;
    sellTotalFees = sellRewardsFee + sellLiquidityFee + sellOperationFee;
}
```

Recommendation

The contract could embody a check for the maximum acceptable value.

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.

Team Update

The team has acknowledged the threat and transferred the contract ownership to a multi-sign mechanism.

BC - Blacklisted Contracts

Criticality	medium
Location	contract.sol#L1480

Description

The contract owner has the authority to stop contracts from transactions. The owner may take advantage of it by calling the `setIsBot` function.

```
require(!_isBot[to] || !_isBot[from], "AltSwitch: To/from address is ignored");
```

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.

Team Update

The team has acknowledged the threat and transferred the contract ownership to a multi-sign mechanism.

Contract Diagnostics

● Critical ● Medium ● Minor

Severity	Code	Description
●	L01	Public Function could be Declared External
●	L05	Unused State Variable
●	L04	Conformance to Solidity Naming Conventions
●	L09	Dead Code Elimination
●	L11	Unnecessary Boolean equality
●	L07	Missing Events Arithmetic

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L1862,L1818,L1668 and 30 more

Description

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

```
process
getAccountAtIndex
size
...
```

Recommendation

Use the external attribute for functions never called from the contract

L05 - Unused State Variable

Criticality	minor
Location	contract.sol#L211

Description

There are segments that contains unused state variable.

```
MAX_INT256
```

Recommendation

Remove unused state variables.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L1773,L1040,L1019 and 14 more

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.

```
_account
RewardsFee
_isBot
...
```

Recommendation

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

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L216,L228,L257 and 7 more

Description

Functions that are not used in the contract, and make the code's size bigger.

```
mul
div
abs
...
```

Recommendation

Remove unused functions.

L11 - Unnecessary Boolean equality

Criticality	minor
Location	contract.sol#L1386,L828

Description

The comparison to boolean constants is redundant. Boolean constants can be used directly and do not need to be compared to true or false.

```
require(bool,string)(isAMMWhitelisted(ammContractAddress) == true,AltSwitch:
setRewardToken:: AMM is not whitelisted!)
userHasCustomRewardToken[holder] == true
```

Recommendation

Remove the equality to the boolean constant.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L1246,L1239,L1202 and 1 more

Description

Detected missing events for critical arithmetic parameters. There are functions that have no event emitted, so it is difficult to track off-chain changes.

```
sellRewardsFee = _rewards
RewardsFee = _rewards
maxSellPercentage = _maxSellPercent
...
```

Recommendation

Emit an event for critical parameter changes.

Contract Functions

Contract	**Type**	**Bases**		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
Address	Library			
	sendValue	Internal	✓	
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_setOwner	Private	✓	
IFactory	Interface			
	createPair	External	✓	-
IPair	Interface			
	getReserves	External		-
	token0	External		-

IRouter	Interface			
	factory	External		-
	WETH	External		-
	addLiquidityETH	External	Payable	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
DividendPayingTokenOptionalInterface	Interface			
	withdrawableDividendOf	External		-
	withdrawnDividendOf	External		-
	accumulativeDividendOf	External		-
DividendPayingTokenInterface	Interface			
	dividendOf	External		-
	distributeDividends	External	Payable	-
	withdrawDividend	External	✓	-
SafeMathInt	Library			
	mul	Internal		
	div	Internal		
	sub	Internal		
	add	Internal		
	abs	Internal		
	toUint256Safe	Internal		
SafeMathUint	Library			

	toInt256Safe	Internal		
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
ERC20	Implementation	Context, IERC20, IERC20Metadata		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_beforeTokenTransfer	Internal	✓	

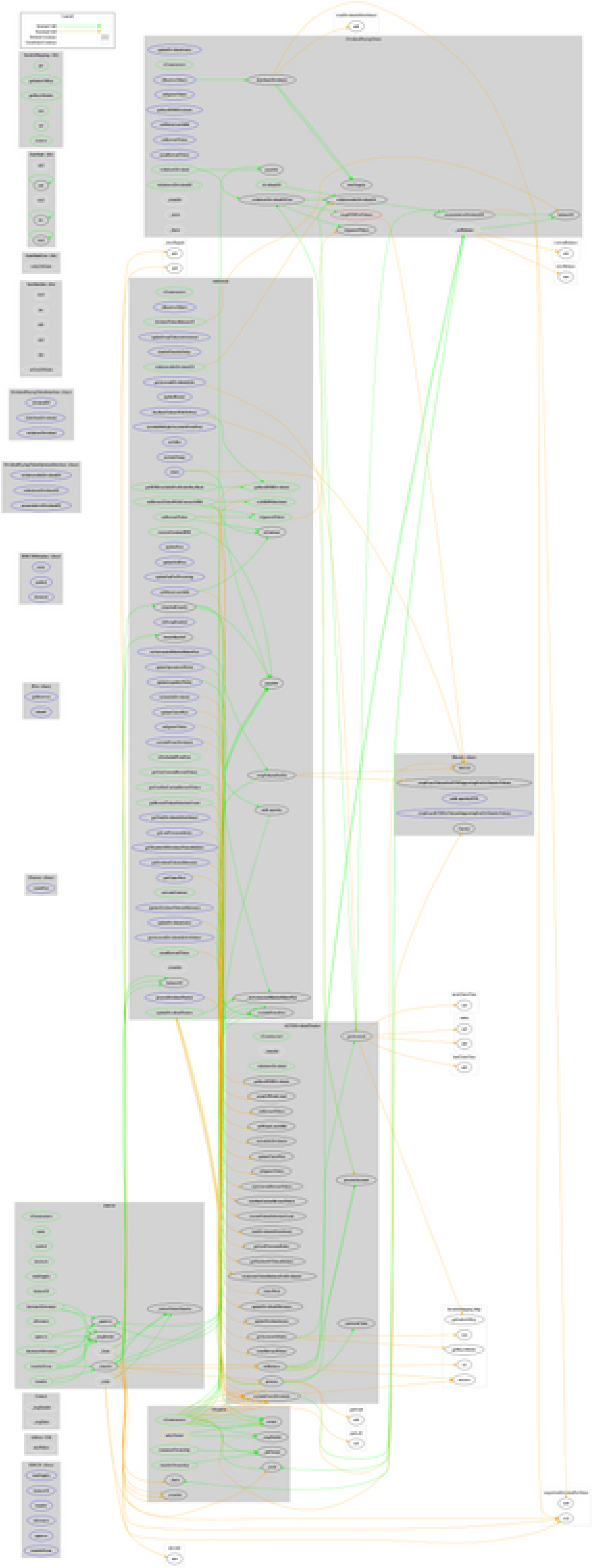
DividendPayingToken	Implementation	ERC20, DividendPayingTokenInterface, DividendPayingTokenOptionalInterface, Ownable		
	updateDividendrouter	External	✓	onlyOwner
	<Constructor>	Public	✓	ERC20
	<Receive Ether>	External	Payable	-
	swapETHForTokens	Private	✓	
	setIgnoreToken	External	✓	onlyOwner
	isIgnoredToken	Public		-
	getRawBNBDividends	External		-
	setWhiteListAMM	External	✓	onlyOwner
	setRewardToken	External	✓	onlyOwner
	unsetRewardToken	External	✓	onlyOwner
	distributeDividends	Public	Payable	-
	withdrawDividend	Public	✓	-
	_withdrawDividendOfUser	Internal	✓	
	dividendOf	Public		-
	withdrawableDividendOf	Public		-
	withdrawnDividendOf	Public		-
	accumulativeDividendOf	Public		-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_setBalance	Internal	✓	
AltSwitch	Implementation	ERC20, Ownable		
	<Constructor>	Public	✓	ERC20
	<Receive Ether>	External	Payable	-
	setWhiteListAMM	External	✓	onlyOwner
	updateSwapTokensAtAmount	External	✓	onlyOwner
	disableTransferDelay	External	✓	onlyOwner

	updateDividendTracker	Public	✓	onlyOwner
	updateDividendTokensMinimum	External	✓	onlyOwner
	updateRouter	External	✓	onlyOwner
	updateDividendrouter	External	✓	onlyOwner
	excludeFromFees	Public	✓	onlyOwner
	excludeMultipleAccountsFromFees	External	✓	onlyOwner
	setIsBot	External	✓	onlyOwner
	setAntiDump	External	✓	onlyOwner
	excludeFromDividends	External	✓	onlyOwner
	includeInDividends	External	✓	onlyOwner
	setAutomatedMarketMakerPair	External	✓	onlyOwner
	updateLiquidityWallet	External	✓	onlyOwner
	updateOperationsWallet	External	✓	onlyOwner
	updateFees	External	✓	onlyOwner
	updateSellFees	External	✓	onlyOwner
	updateGasForProcessing	External	✓	onlyOwner
	updateClaimWait	External	✓	onlyOwner
	setIgnoreToken	External	✓	onlyOwner
	setSwapEnabled	External	✓	onlyOwner
	isAMMWhitelisted	Public		-
	isContract	Internal		
	getUserCurrentRewardToken	Public		-
	getUserHasCustomRewardToken	Public		-
	getRewardTokenSelectionCount	Public		-
	getLastProcessedIndex	External		-
	getNumberOfDividendTokenHolders	External		-
	getDividendTokensMinimum	External		-
	getClaimWait	External		-
	getTotalDividendsDistributed	External		-
	isExcludedFromFees	Public		-
	withdrawableDividendOf	Public		-
	dividendTokenBalanceOf	Public		-
	getAccountDividendsInfo	External		-
	getAccountDividendsInfoAtIndex	External		-
	getRawBNBDividends	Public		-

	getBNBAvailableForHolderBuyBack	Public		-
	isIgnoredToken	Public		-
	setRewardToken	Public	✓	-
	setRewardTokenWithCustomAMM	Public	✓	-
	unsetRewardToken	Public	✓	-
	activateContract	Public	✓	onlyOwner
	buyBackTokensWithNoFees	External	Payable	-
	claim	External	✓	-
	processDividendTracker	External	✓	-
	_setAutomatedMarketMakerPair	Private	✓	
	checkMaxSell	Internal		
	_transfer	Internal	✓	
	swapAndLiquify	Private	✓	
	swapTokensForEth	Private	✓	
	addLiquidity	Private	✓	
	recoverContractBNB	Public	✓	onlyOwner
IterableMapping	Library			
	get	Public		-
	getIndexOfKey	Public		-
	getKeyAtIndex	Public		-
	size	Public		-
	set	Public	✓	-
	remove	Public	✓	-
ALTSDividend Tracker	Implementation	DividendPayingToken		
	<Constructor>	Public	✓	DividendPayingToken
	_transfer	Internal		
	withdrawDividend	Public		-
	excludeFromDividends	External	✓	onlyOwner
	includeInDividends	External	✓	onlyOwner
	updateDividendMinimum	External	✓	onlyOwner
	updateClaimWait	External	✓	onlyOwner

	getLastProcessedIndex	External		-
	getNumberOfTokenHolders	External		-
	getAccount	Public		-
	getAccountAtIndex	Public		-
	canAutoClaim	Private		
	setBalance	External	✓	onlyOwner
	process	Public	✓	-
	processAccount	Public	✓	onlyOwner

Contract Flow



Domain Info

Domain Name	altswitch.io
Registry Domain ID	a3ff63e680e14f5a996a7ef53ca53428-DONUTS
Creation Date	2021-11-29T08:54:02Z
Updated Date	2021-12-04T08:54:07Z
Registry Expiry Date	2022-11-29T08:54:02Z
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 has been created about 2 months before the creation of the audit. It will expire in 11 months.

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

Summary

AltSwitch is aiming to build apps and services for Web 3.0 that will generate revenue and rewards back to its community. The token has a friendly and growing community. There are some functions that can be abused by the owner, like manipulating fees, blacklisting wallets, stopping transactions and stopping users from selling their tokens. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats.

Team Update

The team has acknowledged the threats and transferred the contract ownership to a multi-sign mechanism.

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.

Coinscope 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 Coinscope 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 Coinscope 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 Coinscope team disclaims any liability for the resulting losses.

About Coinscope

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.

Coinscope is aiming to make crypto discoverable and efficient globally. It provides all the essential tools to assist users draw their own conclusions.

The Coinscope.co team

https://www.coinscope.co