



Smart Contract Executive Audit.

August,2021
EggChain Project.

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice and the ISO27001 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 EggChain and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (EggChain) owe no duty of care towards you or any other person, nor does EggChain 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 EggChain hereby excludes all representations, warranties and other terms 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, EggChain hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against EggChain, 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. The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.

Background.

WolfSafePoorPeople, thanks to the affiliation and permission of Alfian the CEO of WSPP, carried out the audit of the contract:

- WSPP:
<https://bscscan.com/address/0x46d502fac9aea7c5bc7b13c8ec9d02378c33d36f#code>

The purpose of this analysis is:

- Ensure that the smart contract functions achieve the objectives expected by the users.
- Identify security breaches that affect the user's CIA in any way.

The information acquired in this document will expose the risk levels of interacting with the smart contract of the WolfSafePoorPeople Plataform.

SmartContract WSPP function list:

*Function Allowance:	
What performs this function?	Allows you to view the amount allowed to spend from WSPP to another contract.
Is it what the user expects?	PASS

*Function BalanceOf:	
What performs this function?	Allows you to view the correct WSPP balance for each user or contract.
Is it what the user expects?	PASS

*Function aAmt:	
What performs this function?	Airdrop amount.
Is it what the user expects?	PASS

*Function viewAirdrop:	
What performs this function?	View the current parameters of the airdrop.
Is it what the user expects?	PASS

*Function newOwner:	
What performs this function?	Shows the candidate for the next contract owner.
Is it what the user expects?	PASS

*Function sEBlock:	
What performs this function?	Block when the exit ends.
Is it what the user expects?	PASS

*Function sCap:	
What performs this function?	Sale capital.
Is it what the user expects?	PASS

*Function owner:	
What performs this function?	Shows the current owner of the contract.
Is it what the user expects?	PASS

*Function aTot:	
What performs this function?	Airdrop counter.
Is it what the user expects?	PASS

*Function sTot:	
What performs this function?	Sale counter.
Is it what the user expects?	PASS

*Function viewSale:	
What performs this function?	Displays the configured parameters of the current sale.
Is it what the user expects?	PASS

*Function sPrice:	
What performs this function?	Sale price.
Is it what the user expects?	PASS

*Function TotalSupply:	
What performs this function?	Displays the WSPP TotalSupply.
Is it what the user expects?	PASS

*Function aSBlock:	
What performs this function?	Airdrop starting block.
Is it what the user expects?	PASS

*Function getAirdrop:	
What performs this function?	Allows you to claim earnings from referrals.
Is it what the user expects?	PASS

*Function startSale:	
What performs this function?	Start the sale with the parameters of: sSBlock sEBlock sChunk sPrice sCap sTot = 0
Is it what the user expects?	PASS

*Function clearETH:	
What performs this function?	Recover the BNB from the contract, can only be called by the owner.
Is it what the user expects?	PASS

*Function tokenSale:	
What performs this function?	Allows the purchase with a referral.
Is it what the user expects?	PASS

*Function startAirdrop:	
What performs this function?	Start the sale with the parameters of: aSBlock aEBlock aAmt aCap aTot = 0
Is it what the user expects?	PASS

*Function transferOwnership:	
What performs this function?	The current owner chooses who will be the new owner of the contract.
Is it what the user expects?	PASS

*Function acceptOwnership:	
What performs this function?	Only the next previously authorized contract owner can call this function to accept the privileges..
Is it what the user expects?	PASS

*Function ApproveAndCall:	
What performs this function?	Approve the spend of tokens to a contract or user and issue the spend at the same time.
Is it what the user expects?	PASS

*Function Transfer:	
What performs this function?	It allows the user to transfer their funds to another wallet.
Is it what the user expects?	PASS

*Function TransferFrom:	
What performs this function?	Allows a contract to transfer funds previously approved by another user.
Is it what the user expects?	PASS

*Function approve:	
What performs this function?	Allows a contract to transfer funds previously approved by another user.
Is it what the user expects?	PASS

Security issues

High Severity Issues:

There are no vulnerabilities that affect the funds deposited by the user or the trading.

Medium Severity Issues:

N/A

Low Severity Issues:

No consensus mechanisms.

Owner Privileges:

Dev wallet is the current owner, without advanced privileges that affect the user in any way.

Tokenomics of WSPP (14/08/2021):

+Maximum Supply: 49,999,999,750,000,000

+Total Burned: 33,478,956,283,175,900

+Locked: 3,300,717,479,560,480

73.54% of Supply is LOCKED in Pool or burned.

+Circulatory Supply: 13,229,999,933,850,000

–WSPP in Liquidity

(Pancakeswap):813,899,053,455,932

–Dev Wallet: 1,363,234,081,050,610

Conclusion:

User funds are invulnerable, the contracts work as they should and do not contain Backdoor to withdraw what is deposited by any user.

EggChain recommends reducing the WSPP amount of the devWallet as it is higher than the circulation in PancakeSwap.

The owner can no longer create WSPP, this is quite positive as it greatly reduces the chances of rug.

FINAL SCORE:      4.5 / 5 EggChain Score.