WP SMART CONTRACTS

BLOCKCHAIN MADE EASY

SECURITY REPORT

WPSmartContracts
Smart Contract Audits

Jun 2022

INTRODUCTION
& Motivation

WPSmartContracts.com is a WordPress plugin in constant evolution, over the years we have set out to develop a product tailored to the needs of our users and a demanding and constantly growing market.

We still consider WPSmartContracts to be a Beta product, and as stated in its terms of use, it is an open source product distributed "as is". But we wanted to take a step forward and offer a product of ever higher quality and efficiency.

In that sense, during the first semester of 2022, we carried out a series of audits on a first batch of smart contracts from WPSmartContracts. These audits included the contracts:

- Ube: Staking
- Almond: Advanced Staking
- Matcha: ERC-721 NFT Marketplace
- Suika: ERC-721 Advanced NFT Marketplace
- Yuzu: ERC-1155 NFT Standard
- Azuki: ERC-1155 NFT Advanced Marketplace
- Ikasumi: ERC-1155 NFT Advanced Token Marketplace

The current report is a summary of the audit results, including technical analysis, corrections, and recommendations.

TEAM & Audits

The audits were carried out by:

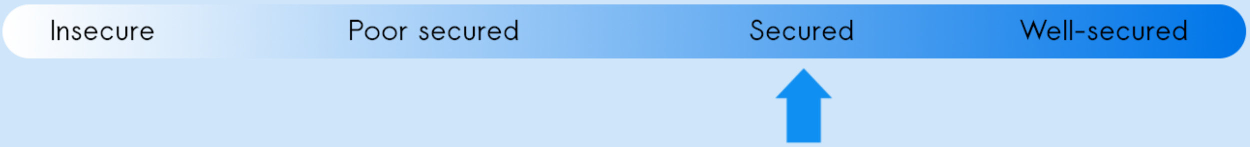
- EtherAuthority.io Team
- Internal team of developers

The purpose of the audit was to ensure that all claimed functions exist and function correctly and to identify any security vulnerabilities that may be present in the smart contract.

The methodology used was divided into four steps: Manual Code Review, Vulnerability Analysis, Documenting Results and Suggested Solutions

AUDIT RESULTS

According to the last audit, WPSmartContracts solidity smart contracts are "Secured"



» **DOWNLOAD THE FULL SECURITY REPORT - #1** (Ube, Almond, Matcha, Suika)
» **DOWNLOAD THE FULL SECURITY REPORT - #2** (Yuzu, Azuki, Ikasumi)

WHY IS THE AUDIT RESULT NOT “WELL SECURED”?

The EtherAuthority team states that to reach the level of "well-secured", the contract must not have any human influence, meaning fully decentralized with no owner control.

For the sake of manageability, we let contract owners decide for themselves whether or not they want to retain ownership of contracts based on their own business rules.

CAN I MAKE MY CONTRACTS FULLY DECENTRALIZED?

Short answer is: it depends.

Please learn more about decentralization in the following link:

» **ABOUT DECENTRALIZATION**

AUDIT DETAILS

UBE
Staking

INITIAL REVIEW

The following table shows the number of findings in the initial review phase classified by severity

Severity	Number	Final status
Critical	0	
High	0	
Medium	0	
Low	2	Acknowledged or fixed
Very low	4	Fixed

CODE CHANGES

The following table shows the findings during the audit and the actions taken to solve it

Severity	Findings	Taken Actions
LOW	**SafeMath is used:** Solidity version above 0.8.0 has in-built integer overflow/underflow protection. So, it is recommended to avoid using safemath.	**Fixed:** Library removed
LOW	**Users may not gain the interest:** In case, the owner does not provide enough allowance, or he does not keep enough token balance into the owner wallet, then users will not receive any interest reward.	**Acknowledged:** We have the following mitigations for this situation: ● A warning before the user ends the stakes in this condition and can contact the owner of the contract / site to solve the problem ● In an emergency, the user can recover the stake funds, and no one else can do it
VERY LOW	**It is helpful to make a view function which outputs if a particular stake is matured or not:** This will be helpful while unstaking, to make sure the premature stake is not withdrawn.	**Acknowledged:** we agree. We prefer to calculate if the stake is mature or not in the interface, subtracting the current timestamp in Javascript by the "from" field of the ledger.
VERY LOW	**Consider using ‘external’ visibility instead of ‘public’.** Although this is not a big problem, it is recommended to use the	**Fixed.** Functions modified as externals

	visibility ‘external’ over ‘public’. It saves some gas as well.	
VERY LOW	**Remove the restriction of max interest rate of 255%.** In most cases an interest rate under 255% is enough, but for some use cases a bigger interest rate is desirable.	**Fixed.** The interest rate limitation was removed changing this variable from uint8 to uint16, now the max interest rate is 65535%
VERY LOW	**Several methods ignore return value by transfer and transferFrom.** The return value of an external transfer/transferFrom call should be checked.	**Fixed.** All call results were checked.

NETWORK UPDATES

UBE Smart Contract was updated on the following networks

» Binance Smart Chain
» Polygon
» Ethereum Classic
» Avalanche
» Fantom

RESULTING CODE

» Click here to see the final code

RECOMMENDATIONS

If you have already deployed an Ube smart contract with previous versions, you do not need to do a new deployment as the changes are low impact.

AUDIT DETAILS

ALMOND
Advanced Staking

INITIAL REVIEW

The following table shows the number of findings in the initial review phase classified by severity

Severity	Number	Final status
Critical	0	
High	0	
Medium	0	
Low	3	Acknowledged or fixed
Very low	4	Fixed

CODE CHANGES

The following table shows the findings during the audit and the actions taken to solve it

Severity	Findings	Taken Actions
LOW	**SafeMath is used:** Solidity version above 0.8.0 has in-built integer overflow/underflow protection. So, it is recommended to avoid using safemath.	**Fixed:** Library removed
LOW	**Users may not gain the interest:** In case, the owner does not provide enough allowance, or he does not keep enough token balance into the owner wallet, then users will not receive any interest reward.	**Acknowledged:** We have the following mitigations for this situation: ● A warning before the user ends the stakes in this condition and can contact the owner of the contract / site to solve the problem ● In an emergency, the user can recover the stake funds, and no one else can do it
LOW	**The method get_gains2 performs a multiplication on the result of a division.** Solidity integer division might truncate, and can sometimes avoid loss of precision.	**Fixed.** Dividend and divisor grouped together
VERY LOW	**It is helpful to make a view function which outputs if a particular stake is matured or not:** This will be helpful while unstaking, to make sure the premature stake is not withdrawn.	**Acknowledged:** we agree. We prefer to calculate if the stake is mature or not in the interface, subtracting the current timestamp in Javascript by the "from" field of the ledger.

VERY LOW	**Consider using 'external' visibility instead of 'public'.** Although this is not a big problem, it is recommended to use the visibility 'external' over 'public'.	**Fixed.** Functions modified as externals
VERY LOW	**Remove the restriction of max interest rate of 255%.** In most cases an interest rate under 255% is enough, but for some use cases a bigger interest rate is desirable.	**Fixed.** The interest rate limitation was removed changing this variable from uint8 to uint16, now the max interest rate is 65535%
VERY LOW	**Several methods ignore return value by transfer and transferFrom.** The return value of an external transfer/transferFrom call should be checked.	**Fixed.** All call results checked.

NETWORK UPDATES

ALMOND Smart Contract was updated on the following networks

» Binance Smart Chain
» Polygon
» Ethereum Classic
» Avalanche
» Fantom

RESULTING CODE

» Click here to see the final code

RECOMMENDATIONS

If you have already deployed an Almond smart contract with previous versions, you do not need to do a new deployment as the changes are low impact.

AUDIT DETAILS

MATCHA
ERC-721 NFT Marketplace

INITIAL REVIEW

The following table shows the number of findings in the initial review phase classified by severity

Severity	Number	Final status
Critical	2	Fixed
High	0	
Medium	0	
Low	3	Acknowledged or fixed
Very low	2	Fixed

CODE CHANGES

The following table shows the findings during the audit and the actions taken to solve it

Severity	Findings	Taken Actions
CRITICAL	Some problems were found in the code. Those findings are confidential.	**Fixed:** all critical issues were fixed.
LOW	**No fractional commission amount possible**: The commission for owner and creators can only be in whole amount and not in fraction. For example, it can only be 1,2,3,etc. It can not be 1.5% or other fractional value.	**Acknowledged:** We consider integer interest values to be fine for the purpose and scope of the smart contract.
LOW	**SafeMath is used:** Solidity version above 0.8.0 has in-built integer overflow/underflow protection. So, it is recommended to avoid using safemath.	**Fixed:** Library removed
LOW	**Older solidity version used.** It is advisable to use the latest solidity version, as many security bugs are fixed in the latest version.	**Fixed.** Recoded in solidity 0.8.2
VERY LOW	**Input validations can be helpful:** The owner can set commission percentages. If the wrong amount has been set by mistake, then it creates discrepancy in the formula.	**Fixed:** we added a condition which specifies the expected percentage variable.

VERY LOW	**Consider using ‘external’ visibility instead of ‘public’.** Although this is not a big problem, it is recommended to use the visibility ‘external’ over ‘public’. It saves some gas as well.	**Fixed.** Functions modified as externals

NETWORK UPDATES

MATCHA Smart Contract was updated on the following networks

» Ethereum
» Binance Smart Chain
» Polygon
» Ethereum Classic
» Avalanche
» Fantom

RESULTING CODE

» Click here to see the final code

RECOMMENDATIONS

Our team did a review and it seems like no smart contracts in mainnet were compromised. But, if you have already deployed a Matcha smart contract with previous versions, you can contact us to verify the ownership of your contract and we will provide you with more information on your particular case.

AUDIT DETAILS

SUIKA
Advanced ERC-721 NFT Marketplace

INITIAL REVIEW

The following table shows the number of findings in the initial review phase classified by severity

Severity	Number	Final status
Critical	0	
High	0	
Medium	0	
Low	2	Acknowledged or fixed
Very low	2	Fixed

CODE CHANGES

The following table shows the findings during the audit and the actions taken to solve it

Severity	Findings	Taken Actions
LOW	**No fractional commission amount possible**: The commission for owner and creators can only be in whole amount and not in fraction. For example, it can only be 1,2,3,etc. It can not be 1.5% or other fractional value.	**Acknowledged:** We consider integer interest values to be fine for the purpose and scope of the smart contract.
LOW	**SafeMath is used:** Solidity version above 0.8.0 has in-built integer overflow/underflow protection. So, it is recommended to avoid using safemath.	**Fixed:** Library removed
VERY LOW	**Input validations can be helpful:** The owner can set commission percentages. If the wrong amount has been set by mistake, then it creates discrepancy in the formula.	**Fixed:** we added a condition which specifies the expected percentage variable.
VERY LOW	**Consider using 'external' visibility instead of 'public'.** Although this is not a big problem, it is recommended to use the visibility 'external' over 'public'. It saves some gas as well.	**Fixed.** Functions modified as externals

NETWORK UPDATES

SUIKA Smart Contract was updated on the following networks

» Binance Smart Chain
» Polygon
» Ethereum Classic
» Avalanche
» Fantom

RESULTING CODE

» Click here to see the final code

RECOMMENDATIONS

If you have already deployed a Suika smart contract with previous versions, you do not need to do a new deployment as the changes are low impact.

AUDIT DETAILS

YUZU
ERC-1155 NFT Standard

INITIAL REVIEW

The following table shows the number of findings in the initial review phase classified by severity

Severity	Number	Final status
Critical	0	
High	0	
Medium	0	
Low	1	Acknowledged
Very low	0	

CODE CHANGES

The following table shows the findings during the audit and the actions taken to solve it

Severity	Findings	Taken Actions
LOW	**Infinite loops possibility**: As array elements will increase, then it will cost more and more gas. And eventually, it will stop all the functionality	**Acknowledged:** Acknowledged, this is part or copied from the OpenZeppelin original logic

NETWORK UPDATES

YUZU Smart Contract was updated on the following networks

» Binance Smart Chain
» Polygon
» Ethereum Classic
» Avalanche
» Fantom

RESULTING CODE

» Click here to see the final code

RECOMMENDATIONS

If you have already deployed a Yuzu smart contract with previous versions, you do not need to do a new deployment as the changes are low impact.

AUDIT DETAILS

AZUKI
ERC-1155 NFT Advanced Marketplace

INITIAL REVIEW

The following table shows the number of findings in the initial review phase classified by severity

Severity	Number	Final status
Critical	0	
High	0	
Medium	2	Fixed
Low	1	Fixed
Very low	0	

CODE CHANGES

The following table shows the findings during the audit and the actions taken to solve it

Severity	Findings	Taken Actions
MEDIUM	**(1) Item creator can bid/buy his own item:** We suggest not allowing the item creator to bid/buy his own item.	**Fixed:** we have added extra validations
MEDIUM	**Commission and Royalty rate can be 100%:** We suggest setting some range below than 100% so that the bid owner will get some token as payment for sure.	**Fixed:** we have added extra validations
LOW	**Critical operation lacks event log:** Missing event log for: cancelSale and cancelBid	**Fixed:** we have added event logs

NETWORK UPDATES

AZUKI is a new Smart Contract and was created on the following networks

» Ethereum
» Binance Smart Chain
» Polygon
» Ethereum Classic
» Avalanche
» Fantom

RESULTING CODE

» Click here to see the final code

AUDIT DETAILS

IKASUMI
ERC-1155 NFT Advanced Token Marketplace

INITIAL REVIEW

The following table shows the number of findings in the initial review phase classified by severity

Severity	Number	Final status
Critical	0	
High	0	
Medium	2	Fixed
Low	1	Fixed
Very low	0	

CODE CHANGES

The following table shows the findings during the audit and the actions taken to solve it

Severity	Findings	Taken Actions
MEDIUM	**(1) Item creator can bid/buy his own item:** We suggest not allowing the item creator to bid/buy his own item.	**Fixed:** we have added extra validations
MEDIUM	**Commission and Royalty rate can be 100%:** We suggest setting some range below than 100% so that the bid owner will get some token as payment for sure.	**Fixed:** we have added extra validations
LOW	**Critical operation lacks event log:** Missing event log for: cancelSale and cancelBid	**Fixed:** we have added event logs

NETWORK UPDATES

AZUKI is a new Smart Contract and was created on the following networks

» Ethereum
» Binance Smart Chain
» Polygon
» Ethereum Classic
» Avalanche
» Fantom

RESULTING CODE

» Click here to see the final code

Conclusions

All audit findings have been properly fixed to ensure smooth operation and greater transparency for our users.

As usual, we encourage our users to do their own research and run their own tests and audits before using this product in production.

This is a step forward to offer a product of increasingly higher quality and efficiency.

Stay informed and stay safe.

WPSMARTCONTRACTS.COM TEAM

REFERENCES

EtherAuthority.io
Audit Report

» **DOWNLOAD THE FULL SECURITY REPORT - #1** (Ube, Almond, Matcha, Suika)

» **DOWNLOAD THE FULL SECURITY REPORT - #2** (Yuzu, Azuki, Ikasumi)