# Worthpad Security Audit Report


Worthpad smart contract security audit, conducted by the Callisto Network Security Department during December 2021.

## Worthpad Token Security Audit Report

*Are Your Funds Safe?*

### Summary

Worthpad (https://github.com/worthpad/worth) smart contract security audit report performed by Callisto Security Audit Department (https://github.com/EthereumCommonwealth/Auditing).

Worthpad ecosystem is powered by the $WORTH Token.

https://worthpad.medium.com/worth-token-the-fuel-that-powers-theworthpad-ecosystem-fe89b9266e33 (https://worthpad.medium.com/worth-token-the-fuel-that-powers-theworthpad-ecosystem-fe89b9266e33)

- Website: https://worthpad.io (https://worthpad.io)
- Twitter: https://twitter.com/worthpad (https://twitter.com/worthpad)

### Platform

Binance Smart Chain.

## 1. In scope

Commit 71760542a40e580ad6c0c57c5ec5798072c0a3b0
(https://github.com/worthpad/worth/tree/71760542a40e580ad6c0c57c5ec5798072c0a3b0)

- WorthToken.sol
- WorthTokenSale.sol
- WorthTokenTimeLock.sol

## 1.1 Excluded

OpenZeppelin standard imports were excluded from the audit.

## 2. Findings

In total, **1 issues** were reported including:

- 0 high severity issues.
- 0 medium severity issue.
- 1 low severity issue.

In total, **10 notes** were reported, including:

- 2 notes.
- 8 owner privileges.

No critical security issues were found.

## 2.1 Owner Privileges

**Description:**

`WorthToken` contract owner has rights to:

1. Exclude/include any account from/in the fee.
2. Set Worth DVC Fund fee percentage in range 1% – 10%.
3. Set liquidity fee percentage in range 1% – 10%.
4. Change the maximal amount per transaction from 0 to 100,000,000 tokens.
5. Enable or disable adding liquidity to pool, using function `setSwapAndLiquifyEnabled` .

`WorthTokenSale` contract owner has rights to:

1. Add users to whitelist and set maximum allocation amount (in USD).
2. Close tokens sale calling function endSale() (https://github.com/worthpad/worth/blob/71760542a40e580ad6c0c57c5ec5798072c0a3b0/WorthTokenSale.sol#L217-L221). Without ending sale users could not claim bought tokens.
3. Withdraw all tokens from contract using function withdrawTokens (https://github.com/worthpad/worth/blob/71760542a40e580ad6c0c57c5ec5798072c0a3b0/WorthTokenSale.sol#L274-L281) include unclaimed users tokens.

## 2.2 allDepositIds is not necessary

**Severity: note.**

**Description:**

The allDepositIds (https://github.com/worthpad/worth/blob/71760542a40e580ad6c0c57c5ec5798072c0a3b0/WorthTokenTimeLock.sol#L59) array contain sequence of `id` from 1 to `depositId`. So all deposits Ids is below or equal to `depositId`.

## 2.3 The Hard cap may be exceeded

**Severity: note.**

**Description**

The Hard cap is checked before adding the amount that the user sends to exchangeUSDTForToken (https://github.com/worthpad/worth/blob/71760542a40e580ad6c0c57c5ec5798072c0a3b0/WorthTokenSale.sol#L161-L164) and exchangeBUSDForToken (https://github.com/worthpad/worth/blob/71760542a40e580ad6c0c57c5ec5798072c0a3b0/WorthTokenSale.sol#L185-L188). It may cause exceed Hard cap if a user sends a bigger amount than left to reach the hard cap.

## 2.4 The owner can withdraw the user's unclaimed tokens

**Severity: low.**

**Description**

The function withdrawTokens (https://github.com/worthpad/worth/blob/71760542a40e580ad6c0c57c5ec5798072c0a3b0/WorthTokenSale.sol#L274-L281) allow the contract owner to withdraw the entire balance of the contract, including tokens that users bought but did not claim yet.

**Recommendation**

Create variable `unclaimedTokens` and add to it amount tokens when user buy it and subtract tokens when user claims it.

In the function `withdrawTokens` withdraw `balance - unclaimedTokens` instead of the entire balance.

## 3. Security practices

- [x] **Open-source contact.**
- [ ] **The contract should pass a bug bounty after the completion of the security audit.**
- [ ] **Public testing.**
- [ ] **Automated anomaly detection systems. – NOT IMPLEMENTED. A simple anomaly detection algorithm is recommended to be implemented to detect behavior that is atypical compared to normal for this contract. For instance, the contract must halt deposits in case a large amount is being withdrawn in a short period of time until the owner or the community of the contract approves further operations.**
- [ ] **Multisig owner account.**

☐ **Standard ERC20-related issues. – NOT IMPLEMENTED. It is known that every contract can potentially receive an unintended ERC20-token deposit without the ability to reject it even if the contract is not intended to receive or hold tokens. As a result, it is recommended to implement a function that will allow extracting any arbitrary number of tokens from the contract.**

☐ **Crosschain address collisions. ETH, ETC, CLO, etc. It is possible that a transaction can be sent to the address of your contract at another chain (as a result of a user mistake or some software fault). It is recommended that you deploy a "mock contract" that would allow you to withdraw any tokens from that address or prevent any funds deposits. Note that you can reject transactions of native token deposited, but you can not reject the deposits of ERC20 tokens. You can use this source code as a mock contract: extractor contract source code (https://github.com/EthereumCommonwealth/GNT-emergency-extractor-contract/blob/master/extractor.sol). The address of a new contract deployed using `CREATE (0xf0)` opcode is assigned following this scheme `keccak256(rlp([sender, nonce]))`. Therefore you need to use the same address that was originally used at the main chain to deploy the mock contract at a transaction with the `nonce` that matches that on the original chain.** *Example: If you have deployed your main contract with address 0x010101 at your 2021th transaction then you need to increase your nonce of 0x010101 address to 2020 at the chain where your mock contract will be deployed. Then you can deploy your mock contract with your 2021th transaction, and it will receive the same address as your mainnet contract.*

## 4. Conclusion

The audited smart contract can be deployed. Only low severity issues were found during the audit.

Pay attention to `WorthTokenSale` contract owner rights that may hurt users.

It is recommended to adhere to the security practices described in pt. 4 of this report to ensure the contract's operability and prevent any issues that are not directly related to the code of this smart contract.

## Appendix

Smart Contract Audits by Callisto Network. (https://callisto.network/smart-contract-audit/)

### Miscellaneous

Why Audit Smart Contracts? (https://callisto.network/why-audit-smart-contracts/)

Our Most Popular Audit Reports. (https://callisto.network/security-audits/)

---

**Trust the Blockchain, Audit the Smart Contracts.**

---

*Follow Callisto's Security Department on Twitter (https://twitter.com/Callisto_Audits) to get our latest news and updates!*

Published on **December 21, 2021**

(https://callisto.network/orthpad-security-audit-report/)

Security Audits (https://callisto.network/tag/security-audits/)

Previous post (https://callisto.network/trust-wallet-token-security-audit-report/) Next post

## Callisto Network LTD

71-75 Shelton Street
London, Greater London
United Kingdom, WC2H 9JQ

## Join Our Community

(https://t.me/CallistoNet) (https://twitter.com/CallistoSupport) (https://reddit.com/r/CallistoCrypto) (https://www.youtube.com/channel/UC1WMae32v_eJ8qOtLQqM26Q)

(https://www.instagram.com/callisto.network/) (https://www.facebook.com/callistonetwork) (https://www.linkedin.com/company/callisto-network/) (https://t.co/DAWunSR1tm)

## Resources

FAQ (https://callisto.network/faq/)

Timeline (https://callisto.network/timeline/)

Airdrop (https://callisto.network/callisto-airdrop/)

Community Guidelines (https://callisto.network/community-guidelines/)

## Callisto

Partners (https://callisto.network/partners/)

Our GitHub repositories (https://github.com/EthereumCommonwealth)

Media Kit (https://github.com/EthereumCommonwealth/Callisto-Media-Kit)

Contact us (https://callisto.network/contact-us/)

Want to sell your CLO coins OTC? (mailto:vladimir.vencalek@invictussolutions.cz)