PALADIN
BLOCKCHAIN SECURITY

Smart Contract Security Assessment

Final Report

For Lootex

28 December 2021

paladinsec.co

info@paladinsec.co

Table of Contents

Disclaimer

Paladin Blockchain Security ("Paladin") has conducted an independent audit to verify the integrity of and highlight any vulnerabilities or errors, intentional or unintentional, that may be present in the codes that were provided for the scope of this audit. This audit report does not constitute agreement, acceptance or advocation for the Project that was audited, and users relying on this audit report should not consider this as having any merit for financial advice in any shape, form or nature. The contracts audited do not account for any economic developments that may be pursued by the Project in question, and that the veracity of the findings thus presented in this report relate solely to the proficiency, competence, aptitude and discretion of our independent auditors, who make no guarantees nor assurance that the contracts are completely free of exploits, bugs, vulnerabilities or deprecation of technologies. Further, this audit report shall not be disclosed nor transmitted to any persons or parties on any objective, goal or justification without due written assent, acquiescence or approval by Paladin.

All information provided in this report does not constitute financial or investment advice, nor should it be used to signal that any persons reading this report should invest their funds without sufficient individual due diligence regardless of the findings presented in this report. Information is provided 'as is', and Paladin is under no covenant to the completeness, accuracy or solidity of the contracts audited. In no event will Paladin or its partners, employees, agents or parties related to the provision of this audit report be liable to any parties for, or lack thereof, decisions and/or actions with regards to the information provided in this audit report.

Cryptocurrencies and any technologies by extension directly or indirectly related to cryptocurrencies are highly volatile and speculative by nature. All reasonable due diligence and safeguards may yet be insufficient, and users should exercise considerable caution when participating in any shape or form in this nascent industry.

The audit report has made all reasonable attempts to provide clear and articulate recommendations to the Project team with respect to the rectification, amendment and/or revision of any highlighted issues, vulnerabilities or exploits within the contracts provided. It is the sole responsibility of the Project team to sufficiently test and perform checks, ensuring that the contracts are functioning as intended, specifically that the functions therein contained within said contracts have the desired intended effects, functionalities and outcomes of the Project team.

Paladin retains full rights over all intellectual property (including expertise and new attack or exploit vectors) discovered during the audit process. Paladin is therefore allowed and expected to re-use this knowledge in subsequent audits and to inform existing projects that may have similar vulnerabilities. Paladin may, at its discretion, claim bug bounties from third-parties while doing so.

1 Overview

This report has been prepared for Lootex on the Ethereum network. Paladin provides a user-centred examination of the smart contracts to look for vulnerabilities, logic errors or other issues from both an internal and external perspective.

1.1 Summary

Project Name	Lootex
URL	https://lootex.io/
Platform	Ethereum
Language	Solidity

1.2 Contracts Assessed

Name	Contract	Live Code Match
`LootToken`	`0x721a1b990699ee9d90b6327faad0a3e840ae8335`	MATCH
`Vesting`	`0x77Df100ccF99CD10D7d5edc4EAc8027b57259b33`	MATCH

1.3 Findings Summary

Severity	Found	Resolved	Partially Resolved	Acknowledged (no change made)
High	1	1	-	-
Medium	1	1	-	-
Low	1	0	-	1
Informational	4	2	-	2
Total	**7**	**4**	**-**	**3**

Classification of Issues

Severity	Description
High	Exploits, vulnerabilities or errors that will certainly or probabilistically lead towards loss of funds, control, or impairment of the contract and its functions. Issues under this classification are recommended to be fixed with utmost urgency.
Medium	Bugs or issues with that may be subject to exploit, though their impact is somewhat limited. Issues under this classification are recommended to be fixed as soon as possible.
Low	Effects are minimal in isolation and do not pose a significant danger to the project or its users. Issues under this classification are recommended to be fixed nonetheless.
Informational	Consistency, syntax or style best practices. Generally pose a negligible level of risk, if any.

1.3.1 Loot Token

ID	Severity	Summary	Status
01	HIGH	Owner of token contract can arbitrarily mint tokens	RESOLVED
02	INFO	Amount minted in constructor has too many digits	RESOLVED

1.3.2 Token Vesting

ID	Severity	Summary	Status
03	MEDIUM	Owner can withdraw all vesting tokens from contract	RESOLVED
04	LOW	Lack of maximum duration check	ACKNOWLEDGED
05	INFO	Start time of vesting schedule can be before time of registration	ACKNOWLEDGED
06	INFO	Beneficiary can be a smart contract which can tokenize timelocked tokens for selling	ACKNOWLEDGED
07	INFO	Debugging statements can be removed from contract code	RESOLVED

2 Findings

2.1 Loot Token

The LOOT token is an ERC20 token with no maximum supply cap. On deployment, 100,000,000 tokens will be minted to the deployer.

Tokens can be burned by an address, or on behalf of an address, provided the address calling the burnFrom has sufficient allowance granted by the address it is burning from.

2.1.1 Token Overview

Address	0x721a1b990699ee9d90b6327faad0a3e840ae8335
Token Supply	100,000,000
Decimal Places	18
Transfer Max Size	No maximum
Transfer Min Size	No minimum
Transfer Fees	None
Pre-mints	100,000,000

2.1.2 Privileged Roles

The following functions can be called by the owner of the contract:

- `transferOwnership`
- `renounceOwnership`
- `mint`

2.1.3 Issues & Recommendations

Issue #01	Owner of token contract can arbitrarily mint tokens
Severity	HIGH SEVERITY
Description	The owner of the token contract is able to mint an unlimited supply of tokens. The ability to do so can greatly impact and shape the tokenomics of the project.
Recommendation	Clarify if there should be a maximum supply cap, and if the owner should be able to freely mint tokens without any restrictions. If there is no maximum supply cap, the owner address should be set to a multisig solution such as Gnosis Safe, to prevent minting and selling of tokens in the case of a key compromise. If there is to be a maximum supply cap, consider minting the full supply initially on deployment, and removing the mint function.
Resolution	RESOLVED The `mint` function has been removed. The maximum supply is now the initial minted supply.

Issue #02	Amount minted in constructor has too many digits
Severity	INFORMATIONAL
Location	Line 11 `_mint(msg.sender, 1000000000 * 10 ** decimals());`
Description	Literals with many digits are difficult to read and review.
Recommendation	Consider changing the amount to something more readable such as `1_000_000_000`.
Resolution	RESOLVED The literal is now more readable.

2.2 Token Vesting

The token vesting contract will hold tokens that are pending to be vested, and will release its token balance gradually like a typical vesting scheme with a cliff and vesting period.

The owner of the contract can register a vesting schedule for a beneficiary. Anyone can help the beneficiary claim tokens that are ready to be released, which will send the tokens to the beneficiary's account. The owner can revoke a vesting schedule provided it was marked as revocable in the registration. The owner can also withdraw the entire vesting token balance in this contract.

2.2.1 Privileged Roles

The following functions can be called by the owner of the contract:

- `transferOwnership`
- `renounceOwnership`
- `register`
- `withdraw`
- `revoke`

2.2.2 Issues & Recommendations

Issue #03	Owner can withdraw all vesting tokens from contract
Severity	MEDIUM SEVERITY
Description	The owner can withdraw all vesting tokens by using the `withdraw` function. This will drain the entire token balance and cause all claims to fail due to the lack of sufficient tokens. In case the owner's private key or seed phrase is compromised, the entire vesting token balance can be stolen.
Recommendation	Considering that this contract will hold a lot of tokens for vesting, consider if it is really necessary to allow the owner to withdraw all the tokens in the contract. If it is required, consider making the owner a multisig. Else, consider removing the `withdraw` function.
Resolution	RESOLVED The `withdraw` function has been removed.

Issue #04	Lack of maximum duration check
Severity	LOW SEVERITY
Description	The lack of a maximum duration check when registering a vesting schedule can result in an extremely long vesting period due to errors such as adding an additional 0 at the end of the duration supplied, or misinterpretation of the duration value being milliseconds instead of seconds.
Recommendation	Add a reasonable maximum duration check in the `register` function.
Resolution	ACKNOWLEDGED The Lootex team has understood the possible risks with this issue and will take extra caution when operating with the contract as a privileged role.

Issue #05	Start time of vesting schedule can be before time of registration
Severity	INFORMATIONAL

Location	Line 174
	`require(`
	`  _start.add(_duration) > block.timestamp,`
	`  "TokenVesting: final time is before current time"`
	`);`

Description	During registration, only the end time is checked to ensure that it is greater than the current time. The start time, however, is not checked.
Recommendation	Confirm if the start time should be at least greater than or equal to the time of registration.
Resolution	ACKNOWLEDGED The Lootex team has understood the possible risks with this issue and will take extra caution when operating with the contract as a privileged role.

Issue #06	Beneficiary can be a smart contract which can tokenize timelocked tokens for selling
Severity	INFORMATIONAL
Description	As mentioned in the following blog post, it would be possible for beneficiaries to trustlessly sell their timelocked tokens without waiting for the timelock to expire. https://blog.openzeppelin.com/bypassing-smart-contract-timelocks/
Recommendation	It is recommended for the beneficiary to sign a non arbitrary message, and to check that the address derived from the signature using ecrecover tallies with the beneficiary address.
Resolution	ACKNOWLEDGED The Lootex team has understood the possible risks with this issue and will take extra caution when operating with the contract as a privileged role.

Issue #07	Debugging statements can be removed from contract code
Severity	INFORMATIONAL
Location	Line 58 `console.log("vestingToken: ", token);`
Description	The above `console.log` statement should be removed from the contract code before deployment.
Recommendation	Consider removing the logging statement.
Resolution	RESOLVED The logging statement has been removed.


PALADIN
BLOCKCHAIN SECURITY