



Openverse Network Security Assessment

CertiK Assessed on Jan 5th, 2025





CertiK Assessed on Jan 5th, 2025

Openverse Network

The security assessment was prepared by CertiK, the leader in Web3.0 security.

Executive Summary

TYPES

DeFi

ECOSYSTEM

Openverse(BTG)

METHODS

Manual Review, Static Analysis, Testnet Deployment

LANGUAGE

Rust

TIMELINE

Delivered on 01/05/2025

KEY COMPONENTS

N/A

CODEBASE

[base](#)

View All in Codebase Page

COMMITTS

[485019a2c592f7d3771004ad16f73874183c38ea](#)

[4d909870149c5f5409cca40c0cded294ca7b3b2e](#)

View All in Codebase Page

Vulnerability Summary



4

Total Findings

1

Resolved

0

Mitigated

0

Partially Resolved

3

Acknowledged

0

Declined

0 Critical

Critical risks are those that impact the safe functioning of a platform and must be addressed before launch. Users should not invest in any project with outstanding critical risks.

1 Major

1 Acknowledged



Major risks can include centralization issues and logical errors. Under specific circumstances, these major risks can lead to loss of funds and/or control of the project.

0 Medium

Medium risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform.

1 Minor

1 Resolved



Minor risks can be any of the above, but on a smaller scale. They generally do not compromise the overall integrity of the project, but they may be less efficient than other solutions.

2 Informational

2 Acknowledged



Informational errors are often recommendations to improve the style of the code or certain operations to fall within industry best practices. They usually do not affect the overall functioning of the code.

TABLE OF CONTENTS | OPENVERSE NETWORK

| Summary

[Executive Summary](#)

[Vulnerability Summary](#)

[Codebase](#)

[Audit Scope](#)

[Approach & Methods](#)

| Review Notes

[System Overview](#)

[External Dependencies](#)

[Privileged Functions](#)

| Findings

[LIB-03 : Centralized Contract Upgradability](#)

[LIB-04 : Missing Validation if the given account already claimed](#)

[LIB-01 : Concerns on Arbitrary Locking Period and Mint](#)

[LIB-02 : Potential Missing Account Closure in Unlock Function](#)

| Appendix

| Disclaimer

CODEBASE | OPENVERSE NETWORK

Repository

base


Commit

485019a2c592f7d3771004ad16f73874183c38ea

4d909870149c5f5409cca40c0cded294ca7b3b2e

AUDIT SCOPE | OPENVERSE NETWORK

1 file audited ● 1 file without findings

ID	Repo	File	SHA256 Checksum
● LIS	openlab- openos/open- token-protocol	 programs/btg-locking-period/src/lib.r s	e0bb53d1c9725fd970feff6d4e6031afd15138d e40122c4e0f22a50623295aa0

APPROACH & METHODS | OPENVERSE NETWORK

This report has been prepared for Openverse to discover issues and vulnerabilities in the source code of the Openverse Network project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Manual Review, Static Analysis, and Testnet Deployment techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

The security assessment resulted in findings that ranged from critical to informational. We recommend addressing these findings to ensure a high level of security standards and industry practices. We suggest recommendations that could better serve the project from the security perspective:

- Testing the smart contracts against both common and uncommon attack vectors;
- Enhance general coding practices for better structures of source codes;
- Add enough unit tests to cover the possible use cases;
- Provide more comments per each function for readability, especially contracts that are verified in public;
- Provide more transparency on privileged activities once the protocol is live.

REVIEW NOTES | OPENVERSE NETWORK

System Overview

This audit concerns the implementation of "btg-locking-period" program.

External Dependencies

The project mainly contains the following dependencies:

Dependency
anchor-lang
spl-token
anchor-spl

It should also be noted here that the code dependencies are in active development in the current auditing version and some of the keywords/functionality may be deprecated in a newer version. It is necessary to keep the dependencies up-to-date to avoid potential vulnerabilities.

The on-chain program can be upgradeable after the initial deployment due to Openverse's features. Also, based on the unique rent mechanism in Openverse, the balance in the account should be carefully set.

We assume these dependencies are valid and non-vulnerable factors and implement proper logic to collaborate with the current project.

Privileged Functions

The Openverse platform allows for the possibility of upgrading its programs, with the default upgrade authority being the entity responsible for deployment. In situations where the program has upgradability features and the account of the upgrade authority becomes compromised, there is the potential for an unauthorized and malicious update to the program.

FINDINGS | OPENVERSE NETWORK



4

Total Findings

0

Critical

1

Major

0

Medium

1

Minor

2

Informational

This report has been prepared to discover issues and vulnerabilities for Openverse Network. Through this audit, we have uncovered 4 issues ranging from different severity levels. Utilizing the techniques of Manual Review, Static Analysis & Testnet Deployment to complement rigorous manual code reviews, we discovered the following findings:

ID	Title	Category	Severity	Status
LIB-03	Centralized Contract Upgradability	Centralization	Major	● Acknowledged
LIB-04	Missing Validation If The Given Account Already Claimed	Logical Issue	Minor	● Resolved
LIB-01	Concerns On Arbitrary Locking Period And Mint	Design Issue	Informational	● Acknowledged
LIB-02	Potential Missing Account Closure In Unlock Function	Design Issue	Informational	● Acknowledged

LIB-03 | CENTRALIZED CONTRACT UPGRADABILITY

Category	Severity	Location	Status
Centralization	● Major	lib.rs (485019a): 1	● Acknowledged

Description

The Openverse platform allows for the possibility of upgrading its programs, with the default upgrade authority being the entity responsible for deployment. In situations where the program has upgradability features and the account of the upgrade authority becomes compromised, there is the potential for an unauthorized and malicious update to the program.

Recommendation

The risk describes the current project design and potentially makes iterations to improve in the security operation and level of decentralization, which in most cases cannot be resolved entirely at the present stage. We advise the client to carefully manage the privileged account's private key to avoid any potential risks of being hacked. In general, we strongly recommend centralized privileges or roles in the protocol be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., multisignature wallets.

Indicatively, here are some feasible suggestions that would also mitigate the potential risk at a different level in terms of short-term, long-term and permanent:

Short Term:

Timelock and Multi sign (2/3, 3/5) combination *mitigate* by delaying the sensitive operation and avoiding a single point of key management failure.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
AND
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key compromised;
AND
- A medium/blog link for sharing the timelock contract and multi-signers addresses information with the public audience.

Long Term:

Timelock and DAO, the combination, *mitigate* by applying decentralization and transparency.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
AND

- Introduction of a DAO/governance/voting module to increase transparency and user involvement.
- AND
- A medium/blog link for sharing the timelock contract, multi-signers addresses, and DAO information with the public audience.

Permanent:

Renouncing the ownership or removing the function can be considered *fully resolved*.

- Renounce the ownership and never claim back the privileged roles.
- OR
- Remove the risky functionality.

I Alleviation

[Openverse Team, 01/05/2025]: The team acknowledged this finding.

[CertiK, 01/05/2025]: It strongly encourages the project team to periodically revisit the private key security management of all addresses related to centralized roles.

LIB-04 | MISSING VALIDATION IF THE GIVEN ACCOUNT ALREADY CLAIMED

Category	Severity	Location	Status
Logical Issue	● Minor	lib.rs (485019a): 62	● Resolved

Description

The `unlock` function lets the token owner withdraw the native token based on the recorded amount in the `LockAccount`.

However, if the owner sets the locked amount exactly to the rent-exempt minimum (e.g., 1,566,000 lampports during testing), it becomes possible to call `unlock` twice—leading to the unintended closure of the `LockAccount`.

Proof of Concept

The following test is produced to validate the aforementioned issue, that user may `unlock` twice and the lock account will be closed:

```
let usdcMint: anchor.web3.PublicKey;

before(async () => {
  usdcMint = await createMint(
    provider.connection,
    provider.wallet.payer,
    provider.wallet.publicKey,
    null,
    6 // 6 decimal places
  );
});

it("run unlock - unpexed close", async () => {
  const amount = 1566000; // Example amount in lamports
  const endTime = (new Date().getTime() / 1000) + 6; // 6s from now

  const tx0 = await program.methods.lock(new anchor.BN(amount), new
anchor.BN(endTime)).accounts({
    lockAccount: lockAccount.publicKey,
    mint: usdcMint,
    owner: provider.wallet.publicKey,
    systemProgram: anchor.web3.SystemProgram.programId,
  })
    .signers([lockAccount])
    .rpc();
  console.log(tx0);

  await new Promise(resolve => setTimeout(resolve, 10000));

  const lockAccountInfoOld = await
provider.connection.getAccountInfo(lockAccount.publicKey);
  console.log("****", lockAccountInfoOld);

  const tx = await program.methods.unlock().accounts({
    lockAccount: lockAccount.publicKey,
    owner: provider.wallet.publicKey,
  })
    .rpc();
  console.log(tx);

  const lockAccountInfoNew = await
provider.connection.getAccountInfo(lockAccount.publicKey);
  console.log("**1**", lockAccountInfoNew);

  await new Promise(resolve => setTimeout(resolve, 5000));

  const tx2 = await program.methods.unlock().accounts({
    lockAccount: lockAccount.publicKey,
```

```

    owner: provider.wallet.publicKey,
  })
  .rpc();
  console.log(tx2);

  const lockAccountInfoNew2 = await
provider.connection.getAccountInfo(lockAccount.publicKey);
  console.log("**2**", lockAccountInfoNew2);

  await new Promise(resolve => setTimeout(resolve, 5000));

  const lockAccountInfoNew3 = await
provider.connection.getAccountInfo(lockAccount.publicKey);
  console.log("**3**", lockAccountInfoNew3);

});

```

```

**** {
  data: <Buffer df 40 47 7c ff 56 76 c0 36 95 aa 10 10 66 f1 b0 80 2c 9f f3 f7 bd 37
62 89 9f 3c 58 40 8e 8a cd ba b8 64 c8 de 8a 62 ef be 44 2b 86 e1 49 f0 7b d2 c8 ...
47 more bytes>,
  executable: false,
  lamports: 3132000,
  owner: PublicKey [PublicKey(7389jrSEejuDdFrgwUsk9ZYenowqVFXkErPwAqLhLutT)] {
    _bn: <BN: 59b15e53a9965c27e6f47b1847bdd375c7cc129745e58016fb8dce9a32ee45d0>
  },
  rentEpoch: 18446744073709552000,
  space: 97
}
DRW8jw72tagbMjUdggN6PtZpvYcyjmbB1s3sDmQ1tqBZ4QMqbKmZ3f32My2YfenHgxbvc2vQWYFU44ETUbo
JQx
**1** {
  data: <Buffer df 40 47 7c ff 56 76 c0 36 95 aa 10 10 66 f1 b0 80 2c 9f f3 f7 bd 37
62 89 9f 3c 58 40 8e 8a cd ba b8 64 c8 de 8a 62 ef be 44 2b 86 e1 49 f0 7b d2 c8 ...
47 more bytes>,
  executable: false,
  lamports: 1566000,
  owner: PublicKey [PublicKey(7389jrSEejuDdFrgwUsk9ZYenowqVFXkErPwAqLhLutT)] {
    _bn: <BN: 59b15e53a9965c27e6f47b1847bdd375c7cc129745e58016fb8dce9a32ee45d0>
  },
  rentEpoch: 18446744073709552000,
  space: 97
}
4ATpy55rdnsomego6sACfgAZwa2VzoApQycGdFkF1cDmoy59nJNjX9BHoDfjgH9i48MZSkCkASmnmq9JGTJ4
puGb
**2** null -> the account closed
**3** null -> the account closed

```

Recommendation

Recommend adding validation on the `is_unlocked` field to prevent the second unlock, if the account is not intended to be closed.

Alleviation

[Openverse Team, 01/05/2025]: The team resolved this finding in commit [4d909870149c5f5409cca40c0cded294ca7b3b2e](#) by adding validation on `is_unlocked`.

LIB-01 | CONCERNS ON ARBITRARY LOCKING PERIOD AND MINT

Category	Severity	Location	Status
Design Issue	● Informational	lib.rs (485019a): 24, 27	● Acknowledged

Description

The following concerns on lock account creation have been raised:

- The program allows a user to specify any `end_time` when creating a lock, without restricting how far in the future it can be. As a result, a user could choose an extremely large or short lock duration.
- Additionally, the program references an SPL Mint and checks `mint.is_initialized` but never actually invokes the Token Program (e.g., `token::transfer`). Instead, it uses the System Program to transfer BTG.
- The given mint account is not involved in any other logic; an arbitrary mint account is allowed.

Recommendation

We would like to check with the team that the aforementioned concerns are intended design.

Alleviation

[Openverse Team, 01/05/2025]: The team added limitation on end time in commit

[4d909870149c5f5409cca40c0cded294ca7b3b2e](#) and confirmed the usage of the mint is intended.

LIB-02 | POTENTIAL MISSING ACCOUNT CLOSURE IN UNLOCK FUNCTION

Category	Severity	Location	Status
Design Issue	● Informational	lib.rs (485019a): 120	● Acknowledged

Description

In the current implementation, when tokens are unlocked via the `unlock` instruction, the `lock_account` remains in existence on-chain even though it may be no longer needed. This means:

- The rent (BTG) paid to maintain this account remains locked
- The account continues to occupy chain state
- Users don't receive back their rent payment

However, the account may be kept as a receipt for onchain or offchain dependencies.

Recommendation

We would like to check with the team if this is intended.

Alleviation

[Openverse Team, 01/05/2025]: The team confirmed the lock is intended to be kept.

APPENDIX | OPENVERSE NETWORK

Finding Categories

Categories	Description
Logical Issue	Logical Issue findings indicate general implementation issues related to the program logic.
Centralization	Centralization findings detail the design choices of designating privileged roles or other centralized controls over the code.
Design Issue	Design Issue findings indicate general issues at the design level beyond program logic that are not covered by other finding categories.

Checksum Calculation Method

The "Checksum" field in the "Audit Scope" section is calculated as the SHA-256 (Secure Hash Algorithm 2 with digest size of 256 bits) digest of the content of each file hosted in the listed source repository under the specified commit.

The result is hexadecimal encoded and is the same as the output of the Linux "sha256sum" command against the target file.

DISCLAIMER | CERTIK

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to you ("Customer" or the "Company") in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes, nor may copies be delivered to any other person other than the Company, without CertiK's prior written consent in each instance.

This report is not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts CertiK to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. CertiK's position is that each company and individual are responsible for their own due diligence and continuous security. CertiK's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

The assessment services provided by CertiK is subject to dependencies and under continuing development. You agree that your access and/or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon, multiple layers of third-parties.

ALL SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF ARE PROVIDED "AS IS" AND "AS AVAILABLE" AND WITH ALL FAULTS AND DEFECTS WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, CERTIK HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, CERTIK SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AND ALL WARRANTIES ARISING FROM COURSE OF DEALING, USAGE, OR TRADE PRACTICE. WITHOUT LIMITING THE FOREGOING, CERTIK MAKES NO WARRANTY OF ANY KIND THAT THE SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF, WILL MEET CUSTOMER'S OR ANY OTHER PERSON'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULT, BE COMPATIBLE OR WORK WITH ANY SOFTWARE, SYSTEM, OR OTHER SERVICES, OR BE SECURE, ACCURATE, COMPLETE, FREE OF HARMFUL CODE, OR ERROR-FREE. WITHOUT LIMITATION TO THE FOREGOING, CERTIK PROVIDES NO WARRANTY OR

UNDERTAKING, AND MAKES NO REPRESENTATION OF ANY KIND THAT THE SERVICE WILL MEET CUSTOMER'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULTS, BE COMPATIBLE OR WORK WITH ANY OTHER SOFTWARE, APPLICATIONS, SYSTEMS OR SERVICES, OPERATE WITHOUT INTERRUPTION, MEET ANY PERFORMANCE OR RELIABILITY STANDARDS OR BE ERROR FREE OR THAT ANY ERRORS OR DEFECTS CAN OR WILL BE CORRECTED.

WITHOUT LIMITING THE FOREGOING, NEITHER CERTIK NOR ANY OF CERTIK'S AGENTS MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND, EXPRESS OR IMPLIED AS TO THE ACCURACY, RELIABILITY, OR CURRENCY OF ANY INFORMATION OR CONTENT PROVIDED THROUGH THE SERVICE. CERTIK WILL ASSUME NO LIABILITY OR RESPONSIBILITY FOR (I) ANY ERRORS, MISTAKES, OR INACCURACIES OF CONTENT AND MATERIALS OR FOR ANY LOSS OR DAMAGE OF ANY KIND INCURRED AS A RESULT OF THE USE OF ANY CONTENT, OR (II) ANY PERSONAL INJURY OR PROPERTY DAMAGE, OF ANY NATURE WHATSOEVER, RESULTING FROM CUSTOMER'S ACCESS TO OR USE OF THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS.

ALL THIRD-PARTY MATERIALS ARE PROVIDED "AS IS" AND ANY REPRESENTATION OR WARRANTY OF OR CONCERNING ANY THIRD-PARTY MATERIALS IS STRICTLY BETWEEN CUSTOMER AND THE THIRD-PARTY OWNER OR DISTRIBUTOR OF THE THIRD-PARTY MATERIALS.

THE SERVICES, ASSESSMENT REPORT, AND ANY OTHER MATERIALS HEREUNDER ARE SOLELY PROVIDED TO CUSTOMER AND MAY NOT BE RELIED ON BY ANY OTHER PERSON OR FOR ANY PURPOSE NOT SPECIFICALLY IDENTIFIED IN THIS AGREEMENT, NOR MAY COPIES BE DELIVERED TO, ANY OTHER PERSON WITHOUT CERTIK'S PRIOR WRITTEN CONSENT IN EACH INSTANCE.

NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS.

THE REPRESENTATIONS AND WARRANTIES OF CERTIK CONTAINED IN THIS AGREEMENT ARE SOLELY FOR THE BENEFIT OF CUSTOMER. ACCORDINGLY, NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH REPRESENTATIONS AND WARRANTIES AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH REPRESENTATIONS OR WARRANTIES OR ANY MATTER SUBJECT TO OR RESULTING IN INDEMNIFICATION UNDER THIS AGREEMENT OR OTHERWISE.

FOR AVOIDANCE OF DOUBT, THE SERVICES, INCLUDING ANY ASSOCIATED ASSESSMENT REPORTS OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

Elevating Your Entire **Web3** Journey

Founded in 2017 by leading academics in the field of Computer Science from both Yale and Columbia University, CertiK is a leading blockchain security company that serves to verify the security and correctness of smart contracts and blockchain-based protocols. Through the utilization of our world-class technical expertise, alongside our proprietary, innovative tech, we're able to support the success of our clients with best-in-class security, all whilst realizing our overarching vision; provable trust for all throughout all facets of blockchain.

