Security Assessment

Alyattes

Oct 22nd, 2021

Table of Contents

Summary

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

- 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.

Overview

Project Summary

Project Name	Alyattes
Platform	BSC
Language	Solidity
Codebase	https://bscscan.com/token/0x72690c447aa1ea53042899b7402d10a176819102
Commit	

Audit Summary

Delivery Date	Oct 22, 2021
Audit Methodology	Static Analysis, Manual Review
Key Components	

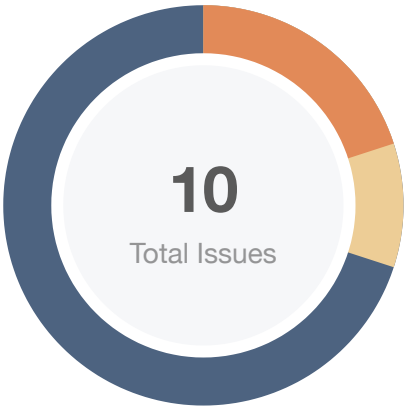
Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Partially Resolved	Resolved
Critical	0	0	0	0	0	0
Major	2	0	0	1	0	1
Medium	0	0	0	0	0	0
Minor	1	0	0	0	0	1
Informational	7	0	0	2	1	4
Discussion	0	0	0	0	0	0

Audit Scope

ID			File	SHA256 Checksum

Findings



Critical	**0** (0.00%)
Major	**2** (20.00%)
Medium	**0** (0.00%)
Minor	**1** (10.00%)
Informational	**7** (70.00%)
Discussion	**0** (0.00%)

ID	Title	Category	Severity	Status
GLOBAL-01	Missing event emitting	Coding Style	Informational	Resolved
GLOBAL-02	Declaration Naming Convention	Coding Style	Informational	Acknowledged
ATC-01	Privileged ownership	**Centralization / Privilege**	**Major**	Resolved
ATC-02	Redundant code	Logical Issue, Gas Optimization	Informational	Partially Resolved
ATC-03	Typos in the contract	Coding Style	Informational	Acknowledged
ATC-04	Variable could be declared as `uint256`	Optimization	Informational	Resolved
ATC-05	Variable could be declared as `constant`	Gas Optimization	Informational	Resolved
ATC-06	Missing approval checks when calling `_transfer()`	Logical Issue	Major	Acknowledged
ATC-07	`_isExcluded` excludes user from both fees and rewards	Logical Issue	Informational	Resolved
ATC-08	Mining excludes you from reflections	Logical Issue	Minor	Resolved

GLOBAL-01 | Missing event emitting

Category	Severity	Location	Status
Coding Style	Informational	Global	Resolved

Description

In contract `AlyaToken`, there are a bunch of functions can change state variables. However, these function do not emit event to pass the changes out of chain.

Recommendation

Recommend emitting events, for all the essential state variables that are possible to be changed during runtime.

Alleviation

[Alyattes Team]:

We are aware about lots of variables but all of them are necessary for an efficient Proof of Active Mining and automated Exclude and Include Operations in advance of Staker's Benefits.

Since the contract has already been published and above described Issue doesn't cause a critical error, it is not necessary to perform any changes.

GLOBAL-02 | Declaration Naming Convention

Category	Severity	Location	Status
Coding Style	Informational	Global	Acknowledged

Description

Code does not conform to the Solidity style guide with regards to its naming convention.

Particularly:

- `camelCase`: Should be applied to function names, argument names, local and state variable names, modifiers
- `UPPER_CASE`: Should be applied to `constant` variables
- `CapWords`: Should be applied to contract names, struct names, event names and enums"

Recommendation

We advise that the linked variable and function names are adjusted to properly conform to Solidity's naming convention.

Alleviation

[Alyattes Team]: Missing Solidity Naming Conventions doesn’t create a risk on Smart Contract or its Functions.

UPPERCASE, camelCase and CapWords are working as planned.

Since the contract has already been published and above described Issue doesn‘t cause a critical error, it is not necessary to perform any changes.

ATC-01 | Privileged ownership

Category	Severity	Location	Status
Centralization / Privilege	Major	projects/alyattes/contracts/AlyaToken.sol: 607~642	Resolved

Description

The owner of contract `AlyaToken` has the permission to:

- `setAsCharityAccount`
- `updateFee`
- `excludeAccount`
- `includeAccount` without obtaining the consensus of the community.

Recommendation

We advise the client to carefully manage the `Owner` account's private key to avoid any potential risks of being hacked. In general, we strongly recommend centralized privileges or roles in the protocol to be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., Multisignature wallets.

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

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key;
- Introduction of a DAO/governance/voting module to increase transparency and user involvement.

Alleviation

[Alyattes Team]: Alya Team is aware of this Privileged Ownership. Owner Wallet is already a Gnosis Multisig Wallet which requires the approval of the CEO, CTO and CFO. (at least 2 of them).

Ownership may be transferred to 0x0000dead after all transactions and processes are settled in the future.

ATC-02 | Redundant code

Category	Severity	Location	Status
Logical Issue, Gas Optimization	Informational	projects/alyattes/contracts/AlyaToken.sol: 963, 962, 919 ~922, 861~865, 901~911, 671~672, 1043~1115	Partially Resolved

Description

1. (L671 - L672) The condition `!_isExcluded[sender] && !_isExcluded[recipient]` is included in `else`.
2. (L901 - L911) The logic in `stakerStatus` can be replaced with its condition.
3. (L861 - L865) The struct `activeMiners` is not needed if it only stores one primitive value.
4. (L919 - L922) Else can be removed from the 2nd return statement.
5. (L1043 - L1115) Memo system already exists in the Binance Smart Chain.
6. (L962 - L963) Remove casting to `uint256` when the given variable is already of type `uint256`.

Recommendation

1. The following code can be removed:

```
    else if (!_isExcluded[sender] && !_isExcluded[recipient]) {
            _transferStandard(sender, recipient, amount);
    }
```

2. The following code can be replaced:

```
    function stakerStatus(address _addr) view public returns(bool){

        if(nStockDetails[_addr]._stocktime == 0)
        {
            return false;
        }
        else
        {
            return true;
        }
    }
```

with this:

```
    function stakerStatus(address _addr) view public returns(bool){
        return nStockDetails[_addr]._stocktime != 0;
    }
```

3. The following code can be replaced:

```
struct activeMiners {
      address bUser;
}

mapping(uint256 => activeMiners[]) aMiners;
```

with this:

```
mapping(uint256 => address[]) aMiners;
```

and all calls of `activeMiners()` can be removed.

4. Remove the `else` wrapping in the following lines of code:

```
        else
        {
            return nStockDetails[_addr]._stockamount;
        }
```

5. We recommend removing parts of the contract that are related to creating/getting memos.
6. We advise to remove the redundant casting to `uint256` on the aforementioned lines to save gas cost associated with it.

Alleviation

[Alyattes Team]:

3. Proof of Active Consensus needs this Code because there could be active stakers who are not signing the Reward Blocks. Recording Active Miners is vital to keep Reward Distrubution fair.

4. „else“ command is necessary because Contract needs a Result if `stakeamount` is not 0.

Generally, Certik's recommendations which reduce some lines on the contract may optimize Transfer Fee’s but this optimization will be about 0,0001% less on each Transfer.

Since the contract has already been published and above described Issues don‘t cause a critical error or make a huge Optimisation on Gas Fee’s, it is not necessary to perform any changes.

ATC-03 | Typos in the contract

Category	Severity	Location	Status
Coding Style	Informational	projects/alyattes/contracts/AlyaToken.sol: 887, 483	Acknowledged

Description

There are several typos in the code.

1. In the following code snippet, `nRewarMod` should be `nRewardMod`.

```
1  uint256 public nRewarMod;
```

2. In the following code snippet, `addressHashs` should be `addressHashes`.

```
1      function addressHashs() view public returns (uint256) {
```

Recommendation

We recommend correcting all typos in the contract.

Alleviation

[Alyattes Team]:

Typos are not causing major issues on functionality of contract and variables may have any random definitions/names.

Since the contract has already been published and above Typos don‘t cause a critical error, it is not necessary to perform any changes.

ATC-04 | Variable could be declared as `uint256`

Category	Severity	Location	Status
Optimization	Informational	projects/alyattes/contracts/AlyaToken.sol: 842	Resolved

Description

Variables `totalminers` could be declared as `uint256` since this state variable is read only for its length and only pushes miners into the array.

Recommendation

We recommend declaring those variables as `uint256` and replace all instances with `totalminers.push(...)` with `totalminers += 1`.

Alleviation

[Alyattes Team]:

Existing Contract reaches same Result as Recommandation.

ATC-05 | Variable could be declared as `constant`

Category	Severity	Location	Status
Gas Optimization	Informational	projects/alyattes/contracts/AlyaToken.sol: 465, 463	Resolved

Description

Variables `_MAX` and `_GRANULARITY` could be declared as `constant` since these state variables are never to be changed.

Recommendation

We recommend declaring those variables as `constant`.

Alleviation

[Alyattes Team]:

Existing Contract reaches same Result as Recommandation.

ATC-06 | Missing approval checks when calling `_transfer()`

Category	Severity	Location	Status
Logical Issue	Major	projects/alyattes/contracts/AlyaToken.sol: 1060, 1054, 954	Acknowledged

Description

In the mining/staking functions that call `_transfer()` where the tokens are going from `msg.sender` to `address(this)`, they do not check the `msg.sender`'s allowance for `address(this)`.

Recommendation

We recommend calling the `_approve` function with the spender being `address(this)` and the sender being `_msgSender()`.

Alleviation

[Alyattes Team]: Because there are 4 different versions under the transfer command implemented, using only `transferFrom` code may cause an error.

Since this issue don't affect contracts functionality, it is not necessary to perform any changes.

ATC-07 | `_isExcluded` excludes user from both fees and rewards

Category	Severity	Location	Status
Logical Issue	Informational	projects/alyattes/contracts/AlyaToken.sol: 660~662	Resolved

Description

User is excluded from both rewards and fees.

Alleviation

[Alyattes Team]:

`_isExcluded` is necessary for the fairness of Reward Distribution due to Team Wallets.

Otherwise Team Wallets are going to be Whales and get most of the Rewards due to their higher holding Amounts.

ALYA is planning to work Basic Users oriented and excluded its own wallets from distribution.

For further Information of Excluded Wallets, we refer to our Whitepaper.

Whitepaper Link : https://github.com/Alyattes/WhitePaper/blob/main/EN.pdf

ATC-08 | Mining excludes you from reflections

Category	Severity	Location	Status
Logical Issue	Minor	projects/alyattes/contracts/AlyaToken.sol: 951	Resolved

Description

Once a non-excluded user successfully calls the `startMining` function, they will be excluded from reflections, even on the tokens they kept before mining.

Alleviation

[Alyattes Team]: After using `_startmining` command, existing Wallet is going to be excluded from all Rewards and Fee's even if it has coins before excluding. First reason for that Excluding is to be able to provide entire Mining Rewards to the Stakers and second reason is providing refunding entire Staked Amount tot he Miners back.

As soon as a Miner use `payback` function to end Mining Period, Stake Wallet is going to be Included again.

Appendix

Finding Categories

Centralization / Privilege

Centralization / Privilege findings refer to either feature logic or implementation of components that act against the nature of decentralization, such as explicit ownership or specialized access roles in combination with a mechanism to relocate funds.

Gas Optimization

Gas Optimization findings do not affect the functionality of the code but generate different, more optimal EVM opcodes resulting in a reduction on the total gas cost of a transaction.

Logical Issue

Logical Issue findings detail a fault in the logic of the linked code, such as an incorrect notion on how block.timestamp works.

Coding Style

Coding Style findings usually do not affect the generated byte-code but rather comment on how to make the codebase more legible and, as a result, easily maintainable.

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

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.

About

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.