CERTIK

Security Assessment

Planet Sandbox

Sept 29th, 2021

Table of Contents

Summary

This report has been prepared for Planet Sandbox to discover issues and vulnerabilities in the source code of the Planet Sandbox 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	Planet Sandbox
Description	ERC20
Platform	Custom
Language	Solidity
Codebase	The client provided the source files in a zip.
Commit	

Audit Summary

Delivery Date	Sept 29, 2021
Audit Methodology	Static Analysis, Manual Review
Key Components	

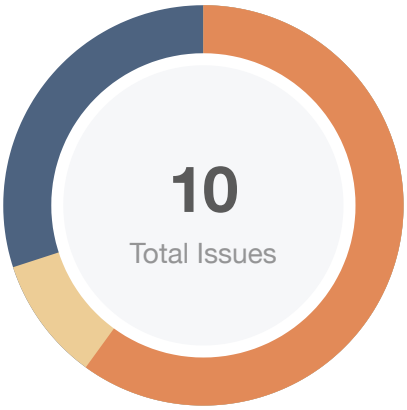
Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Partially Resolved	Resolved
Critical	0	0	0	0	0	0
Major	6	0	0	2	1	3
Medium	0	0	0	0	0	0
Minor	1	0	0	0	0	1
Informational	3	0	0	0	0	3
Discussion	0	0	0	0	0	0

Audit Scope

ID	File	SHA256 Checksum
PSC	PlanetSandbox.sol	fdf1b37ac0d960cd924dcc3e29116a39fd45beec0c4223b753a9df509333d97c
PTC	PulvisToken.sol	dd945bcc77b4c76f62e23e35740173fd2390e47d0e4ef02fbb665827f4540022

Findings



Critical	0 (0.00%)
Major	6 (60.00%)
Medium	0 (0.00%)
Minor	1 (10.00%)
Informational	3 (30.00%)
Discussion	0 (0.00%)

ID	Title	Category	Severity	Status
PSC-01	Centralization Risk	**Centralization / Privilege**	**Major**	Partially Resolved
PSC-02	Centralized Token Holding Position	**Centralization / Privilege**	**Major**	Acknowledged
PSC-03	Owner Can Withdraw Any ERC20 Tokens Held by the Contract	**Centralization / Privilege**	**Major**	Resolved
PSC-04	Possible to Over Vest Token	Volatile Code, Logical Issue	Major	Resolved
PSC-05	Missing Return Value	Coding Style, Inconsistency	Minor	Resolved
PSC-06	Missing Zero Check	Logical Issue, Gas Optimization	Informational	Resolved
PSC-07	No Need to Use SafeMath	Language Specific	Informational	Resolved
PTC-01	Privileged Roles	**Centralization / Privilege**	**Major**	Acknowledged
PTC-02	Centralized Token Holding Position	**Centralization / Privilege**	**Major**	Resolved
PTC-03	No Need to Use SafeMath	Language Specific	Informational	Resolved

PSC-01 | Centralization Risk

Category	Severity	Location	Status
Centralization / Privilege	**Major**	PlanetSandbox.sol (source): 43, 49, 79, 148	Partially Resolved

Description

The role `owner` has the authority over the following function:

- `setTokenVestingTime()` line 43
- `addVestingToken()` line 49
- `revokeVestingToken()` line 79
- `withdrawERC20()` line 148

Any compromise to the `owner` account may allow the hacker to take advantage of this and drastically change the contracts state.

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

[Planet Sandbox]: a `multiSigAccount` is assigned as the role of admin in the updated source code.

PSC-02 | Centralized Token Holding Position

Category	Severity	Location	Status
Centralization / Privilege	Major	PlanetSandbox.sol (source): 37	Acknowledged

Description

The owner of the contract is minted the entire initial token supply at contract deployment.

```
36 constructor(string memory name, string memory symbol) ERC20(name, symbol) {
37     _mint(_msgSender(), INITIAL_SUPPLY);
38 }
```

Recommendation

Once the token goes live, we assume many transactions would involve the wallet unlock of the owner address and the team shall make enough efforts to restrict the access of the private key.

PSC-03 | Owner Can Withdraw Any ERC20 Tokens Held by the Contract

Category	Severity	Location	Status
Centralization / Privilege	**Major**	PlanetSandbox.sol (source): 148~152	Resolved

Description

The Owner is able to call `withdrawERC20()` which would transfer up to the entire balance of any specified ERC20 token held by the contract address.

Recommendation

We recommend removing this function from the contract if possible. If the business logic requires this function, we recommend adequate disclosure of the Owner's privileged access to the community, and carefully manage the Owner'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

[Planet Sandbox]: Function was removed in updated source code.

PSC-04 | Possible to Over Vest Token

Category	Severity	Location	Status
Volatile Code, Logical Issue	Major	PlanetSandbox.sol (source): 60, 143~145	Resolved

Description

In the function `addVestingToken()`, token is not minted and hence `totalSupply()` is not updated. This means that line 60 could pass even if the cumulative `amount` exceeds the `TOTAL_SUPPLY` limit.

```
60  require(totalSupply().add(amount) <= TOTAL_SUPPLY, "PlanetSandbox: Max supply exceeded");
```

If the above scenario is true, a user who calls the `claimVestingToken()` function last might not be able to claim the full amount (or any amount) of token that he or she is eligible for, if previous users have claimed enough tokens such that `totalSupply()` is close to or equal to `TOTAL_SUPPLY` that would cause line 143 to fail.

```
143  require((totalSupply().add(claimableAmount)) <= TOTAL_SUPPLY, "PlanetSandbox: Max supply exceeded");
```

Recommendation

We recommend using a separate `projectedSupply` variable to keep track of the projected future `totalSupply()` of tokens, so that tokens cannot be over vested and unable to be minted for users who claim tokens late.

Alleviation

[Planet Sandbox]: `PROJECTED_SUPPLY` variable included to keep track of intended mint in updated source code.

PSC-05 | Missing Return Value

Category	Severity	Location	Status
Coding Style, Inconsistency	Minor	PlanetSandbox.sol (source): 139	Resolved

Description

The function `claimVestingToken()` is stated return a `uint256`, however not return statement is present and no variable is declared inline to be returned.

Recommendation

We recommend either removing the `returns` declaration or add a `return` statement.

Alleviation

[Planet Sandbox]: `claimableAmount` value now returned in updated source code.

PSC-06 | Missing Zero Check

Category	Severity	Location	Status
Logical Issue, Gas Optimization	Informational	PlanetSandbox.sol (source): 81~84	Resolved

Description

The function `__getVestingClaimableAmount()` returning zero for variable `claimableAmount` would render lines 82 and 84 redundant, wasting gas used in execution.

Recommendation

We recommend the following conditional:

```
81 uint256 claimableAmount = _getVestingClaimableAmount(user);
82 _vestingList[user].isActive = false;
83 if(claimableAmount > 0){
84    require(totalSupply().add(claimableAmount) <= TOTAL_SUPPLY, "PlanetSandbox: Max supply exceeded");
85    _mint(user, claimableAmount);
86 }
```

Alleviation

[Planet Sandbox]: conditional was added in updated source code.

PSC-07 | No Need to Use SafeMath

Category	Severity	Location	Status
Language Specific	Informational	PlanetSandbox.sol (source): 5, 6	Resolved

Description

Solidity version >=0.8.0 includes checked arithmetic operations and underflow/overflow by default, making SafeMath redundant.

Recommendation

We recommend removing the SafeMath library and use standard arithmetic operators to reduce code complexity.

Alleviation

[PlanetSandbox]: import was removed in updated source code.

PTC-01 | Privileged Roles

Category	Severity	Location	Status
Centralization / Privilege	**Major**	PulvisToken.sol (source): 22	Acknowledged

Description

The `_msgSender()` is given the `DEFAULT_ADMIN_ROLE` on contract deployment. This role enables arbitrary assignment of `MINTER_ROLE`, which can mint up to the remaining total supply of tokens.

Recommendation

We advise the client to carefully manage the `DEFAULT_ADMIN_ROLE` and `MINTER_ROLE` 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.

PTC-02 | Centralized Token Holding Position

Category	Severity	Location	Status
Centralization / Privilege	Major	PulvisToken.sol (source): 23	Resolved

Description

The owner of the contract is minted the entire initial token supply at contract deployment.

```
21      constructor() ERC20("PlanetSandbox Pulvis Token", "PULV") {
22          _setupRole(DEFAULT_ADMIN_ROLE, _msgSender());
23          _mint(_msgSender(), INITIAL_SUPPLY);
24      }
```

Recommendation

Once the token goes live, we assume many transactions would involve the wallet unlock of the owner address and the team shall make enough efforts to restrict the access of the private key.

Alleviation

[Planet Sandbox]: the initial minting to contract deployer has been removed in the updated source code.

PTC-03 | No Need to Use SafeMath

Category	Severity	Location	Status
Language Specific	Informational	PulvisToken.sol (source): 6, 7	Resolved

Description

Solidity version >=0.8.0 includes checked arithmetic operations and underflow/overflow by default, making SafeMath redundant.

Recommendation

We recommend removing the SafeMath library and use standard arithmetic operators to reduce code complexity.

Alleviation

[PlanetSandbox]: import was removed in updated source code.

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.

Volatile Code

Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that may result in a vulnerability.

Language Specific

Language Specific findings are issues that would only arise within Solidity, i.e. incorrect usage of private or delete.

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.

Inconsistency

Inconsistency findings refer to functions that should seemingly behave similarly yet contain different code, such as a constructor assignment imposing different require statements on the input variables than a setter function.

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.