Security Assessment

C98Stake

Dec 8th, 2021

Table of Contents

Summary

This report has been prepared for coin98 to discover issues and vulnerabilities in the source code of the C98Stake 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	C98Stake
Description	C98 Stake
Platform	Ethereum
Language	Solidity
Codebase	C98Stake.sol
Commit	

Audit Summary

Delivery Date	Dec 08, 2021
Audit Methodology	Static Analysis, Manual Review
Key Components	

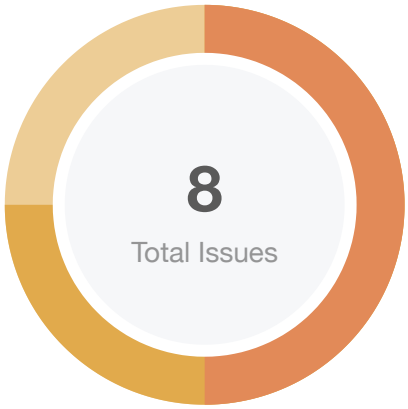
Vulnerability Summary

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

Audit Scope

ID	File	SHA256 Checksum
CSC	C98Stake.sol	7bb10b8eb2ecfbd27f5abcb07bf8e961dd15b6390b4a733ea8f09bfa04a29b93

Findings



Critical	**0** (0.00%)
Major	**4** (50.00%)
Medium	**2** (25.00%)
Minor	**2** (25.00%)
Informational	**0** (0.00%)
Discussion	**0** (0.00%)

ID	Title	Category	Severity	Status
CSC-01	Missing Input Validation	Volatile Code	Minor	Acknowledged
CSC-02	Lack of authorization check in function `renaming()`	Logical Issue	Major	Resolved
CSC-03	Lack of authorization check in function `unstake()`	Logical Issue	Major	Resolved
CSC-04	Unchecked Value of ERC-20 `transfer()`/`transferFrom()` Call	Volatile Code	Medium	Resolved
CSC-05	Centralization Risk	**Centralization / Privilege**	**Major**	Acknowledged
CSC-06	Didn't update `StakeInfos` in function configurePackage()	Logical Issue	Major	Acknowledged
CSC-07	Divide Before Multiply	Mathematical Operations	Minor	Acknowledged
CSC-08	Uncertain Income Source of Reward Token	Logical Issue	Medium	Acknowledged

CSC-01 | Missing Input Validation

Category	Severity	Location	Status
Volatile Code	Minor	projects/C98Stake/C98Stake.sol (32df7cb): 1386~1390	Acknowledged

Description

The given input is missing the check for the non-zero address.

Recommendation

We advise adding the check for the passed-in values to prevent unexpected error.

CSC-02 | Lack of authorization check in function `renaming()`

Category	Severity	Location	Status
Logical Issue	Major	projects/C98Stake/C98Stake.sol (32df7cb): 1600~1602	Resolved

Description

Any user can rename any _tokenId by sending naming_fee.

```
1     function renaming (uint256 _tokenId,string memory _name) public {
2         StakeInfo storage stakeInfo = StakeInfos[_tokenId];
3         require(C98Token.transferFrom(msg.sender, address(this), naming_fee));
```

Recommendation

We advise checking whether this is intended, if not please make sure msg.sender is the owner of _tokenId, and also require only the owner can rename _tokenId's name.

Alleviation

Fixed in provided code.

CSC-03 | Lack of authorization check in function `unstake()`

Category	Severity	Location	Status
Logical Issue	Major	projects/C98Stake/C98Stake.sol (32df7cb): 1618~1622	Resolved

Description

An user can call unstake() successfully even if he is not the owner.

```
1     function unstake(uint256 _tokenId) public {
2         StakeInfo storage stakeInfo = StakeInfos[_tokenId];
3         uint256 _profit = getStakedByTokenId(_tokenId);
4         require(_profit > 0, "Not meet unstake condition");
5         require(ownerOf(_tokenId) == stakeInfo.owner, "Not meet owner condition");
```

Recommendation

We advise checking whether this is intended, if not please make sure msg.sender is the owner of _tokenId.

Alleviation

Fixed in provided code.

CSC-04 | Unchecked Value of ERC-20 `transfer()`/`transferFrom()` Call

Category	Severity	Location	Status
Volatile Code	Medium	projects/C98Stake/C98Stake.sol (32df7cb): 1628~1629, 1650	Resolved

Description

The linked `transfer()`/`transferFrom()` invocations do not check the return value of the function call which should yield a `true` result in case of proper ERC-20 implementation.

Recommendation

As many tokens do not follow the ERC-20 standard faithfully, they may not return a `bool` variable in this function's execution meaning that simply expecting it can cause incompatibility with these types of tokens. Instead, we advise that OpenZeppelin's `SafeERC20.sol` implementation is utilized for interacting with the `transfer()` and `transferFrom()` functions of ERC-20 tokens. The OZ implementation optionally checks for a return value rendering compatible with all ERC-20 token implementations.

Alleviation

Fixed in provided code.

CSC-05 | Centralization Risk

Category	Severity	Location	Status
Centralization / Privilege	Major	projects/C98Stake/C98Stake.sol (32df7cb)	Acknowledged

Description

In the contract `C98Stake.sol`, the role `Owner` has the authority over the following function:

- configureFixedVariable()
- unRegister()
- register()
- setBaseURI()
- withdraw()

Any compromise to the `Owner` account may allow the hacker to take advantage of this, especially the withdraw() function may take over all the funds.

```
1     function withdraw(uint256 _amount) public onlyOwner {
2         require(_amount > 0);
3         require(C98Token.balanceOf(address(this)) >= _amount);
4         C98Token.transfer(msg.sender, _amount);
5     }
```

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.

CSC-06 | Didn't update `StakeInfos` in function configurePackage()

Category	Severity	Location	Status
Logical Issue	Major	projects/C98Stake/C98Stake.sol (32df7cb): 1419~1428	Acknowledged

Description

Function configurePackage() only update `PackageInfos` but not update `StakeInfos`, while the profit calculation still use old `time` and `rate`.

```
    function configurePackage(
        string memory _package,
        uint256 _min,
        uint256 _max,
        uint256 _time,
        uint256 _rate
    )
        public
        onlyOwner()
    {
        require(validPackage(_package), "Package not found");
        require(_min>0 && _max >0 && _min < _max, "Wrong numeric format");

        PackageInfos[_package].min = _min;
        PackageInfos[_package].max = _max;
        PackageInfos[_package].time = _time;
        PackageInfos[_package].rate = _rate;
    }
```

```
getStakedByTokenId() {
...
        } else {
            uint256 calRate = timeStaked < stakeInfo.packageTime ? floating_rate:
stakeInfo.rate;

            uint256 amountProfitBySeconds =
stakeInfo.amount.mul(calRate).div(Percent).div(yearTimestamp);
            return amountProfitBySeconds.mul(timeStaked);
        }
...
```

Recommendation

We advise to also update `stakeInfos` in function configurePackage().

CSC-07 | Divide Before Multiply

Category	Severity	Location	Status
Mathematical Operations	Minor	projects/C98Stake/C98Stake.sol (32df7cb): 1517~1518, 1251	Acknowledged

Description

Mathematical operations in the aforementioned function perform divisions before multiplications. Performing multiplication before division can sometimes avoid loss of precision.

Recommendation

We advise to order multiplication before division.

CSC-08 | Uncertain Income Source of Reward Token

Category	Severity	Location	Status
Logical Issue	Medium	projects/C98Stake/C98Stake.sol (32df7cb)	Acknowledged

Description

According to the implementations of this contract, the users can stake C98 tokens into a package and earn a profit after unstake().

```
    function getStakedByTokenId(uint256 _tokenId) public view returns (uint256) {
        StakeInfo memory stakeInfo = StakeInfos[_tokenId];
        uint256 current = block.timestamp;
        uint256 timeStaked = current - stakeInfo.time;

        if(!stakeInfo.flag || (timeStaked < locked_time)){
            return 0;
        } else {
            uint256 calRate = timeStaked < stakeInfo.packageTime ? floating_rate:
stakeInfo.rate;

            uint256 amountProfitBySeconds =
stakeInfo.amount.mul(calRate).div(Percent).div(yearTimestamp);
            return amountProfitBySeconds.mul(timeStaked);
        }
    }

    function unstake(uint256 _tokenId) public {
        StakeInfo storage stakeInfo = StakeInfos[_tokenId];
        uint256 _profit = getStakedByTokenId(_tokenId);
    ...
```

The profit tokens are all sent from this contract, so the users may not get the full amount of rewards when the balance in this contract is insufficient.

Recommendation

We recommend ensuring every package of this contract has enough profit tokens for users.

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.

Mathematical Operations

Mathematical Operation findings relate to mishandling of math formulas, such as overflows, incorrect operations etc.

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.

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.