CERTIK

Preliminary Comments

Fabwelt

Nov 14th, 2021

Table of Contents

Summary

This report has been prepared for Fabwelt to discover issues and vulnerabilities in the source code of the Fabwelt 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	Fabwelt
Platform	polygon
Language	Solidity
Codebase	https://polygonscan.com/token/0x23e8b6a3f6891254988b84da3738d2bfe5e703b9
Commit	

Audit Summary

Delivery Date	Nov 14, 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	1	1	0	0	0	0
Medium	0	0	0	0	0	0
Minor	0	0	0	0	0	0
Informational	7	7	0	0	0	0
Discussion	0	0	0	0	0	0

Audit Scope

ID	File	SHA256 Checksum
TCK	fabweitToken.sol	3137dcd9f3a3bf7da610f952b2c3b3dada336c3d2814e98b9301591fcd422273

Review Notes

Overview

Fabwelt is a company whose goal is to incorporate blockchain technology such as NFTs and tokens into a variety of games that it develops. The current contract is the implementation of the Fabwelt Token (WELT), which is based on Reflect. The main differences with Reflect are that in addition to the reflect fee, there are two additional types of fees and the owner has the ability to decide all fee rates.

Privileged Functions

In the contracts `Ownable` and `FabweltToken`, the role `_owner` has the authority over the following functions:

- `Ownable.renounceOwnership()`, which renounces the owner role and disables all functions with the `onlyOwner` modifier;
- `Ownable.transferOwnership()`, which transfers the owner role to another address;
- `FabweltToken.excludeAccount()`, which excludes an address's tokens for rate calculations;
- `FabweltToken.includeAccount()`, which includes an excluded address's tokens for rate calculations;
- `FabweltToken.setAsCharityAccount()`, which sets the address for `FeeAddress`;
- `FabweltToken.updateFee()`, which decides the fees for transfers.

To improve the trustworthiness of the project, dynamic runtime updates in the project should be notified to the community. Any plan to invoke the aforementioned functions should be also considered to move to the execution queue of the `Timelock` contract.

Findings

Critical	**0** (0.00%)
Major	**1** (12.50%)
Medium	**0** (0.00%)
Minor	**0** (0.00%)
Informational	**7** (87.50%)
Discussion	**0** (0.00%)

ID	Title	Category	Severity	Status
TCK-01	Centralization Risk	**Centralization / Privilege**	**Major**	Pending
TCK-02	Unlocked Compiler Version	Language Specific	Informational	Pending
TCK-03	Variable can be Declared as Constant	Gas Optimization	Informational	Pending
TCK-04	Lack of Event Emissions for Significant Transactions	Coding Style	Informational	Pending
TCK-05	Redundant Code	Gas Optimization, Coding Style	Informational	Pending
TCK-06	Unused Local Variable	Gas Optimization	Informational	Pending
TCK-07	Check Allowance Before Transfer	Gas Optimization	Informational	Pending
TCK-08	Missing Error Messages	Coding Style	Informational	Pending

TCK-01 | Centralization Risk

Category	Severity	Location	Status
Centralization / Privilege	Major	projects/fabwelt/contracts/fabweitToken.sol (cd81a41): 432, 441, 598, 607, 620, 624	Pending

Description

In the contracts `Ownable` and `FabweltToken`, the role `_owner` has the authority over the following functions:

- `Ownable.renounceOwnership()`, which renounces the owner role and disables all functions with the `onlyOwner` modifier;
- `Ownable.transferOwnership()`, which transfers the owner role to another address;
- `FabweltToken.excludeAccount()`, which excludes an address's tokens for rate calculations;
- `FabweltToken.includeAccount()`, which includes an excluded address's tokens for rate calculations;
- `FabweltToken.setAsCharityAccount()`, which sets the address for `FeeAddress`;
- `FabweltToken.updateFee()`, which decides the fees for transfers.

For the contract deployed at 0x23e8b6a3f6891254988b84da3738d2bfe5e703b9 on Polygon, the `_owner` is 0x63401aac2469bfe676d134571defe64839c35a61, which is an EOA (externally owned account).

Any compromise to the `_owner` account may allow the hacker to take advantage of this and disrupt how the token should operate.

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.

Here are some feasible suggestions that would also mitigate this risk in the short-term and long-term:

- A 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.

TCK-02 | Unlocked Compiler Version

Category	Severity	Location	Status
Language Specific	Informational	projects/fabwelt/contracts/fabweitToken.sol (cd81a41): 11	Pending

Description

The contract has an unlocked compiler version. An unlocked compiler version in the source code of the contract permits the user to compile it at or above a particular version. This, in turn, leads to differences in the generated bytecode between compilations due to differing compiler version numbers. This can lead to ambiguity when debugging as compiler-specific bugs may occur in the codebase that would be hard to identify over a span of multiple compiler versions rather than a specific one.

Recommendation

We advise that the compiler version is instead locked at the lowest version possible that the contract can be compiled at. For example, for version `v0.8.2` the contract should contain the following line:

```
pragma solidity 0.8.2;
```

TCK-03 | Variable can be Declared as Constant

Category	Severity	Location	Status
Gas Optimization	Informational	projects/fabwelt/contracts/fabweitToken.sol (cd81a41): 465, 467	Pending

Description

The values of `FabweltToken._GRANULARITY` and `FabweltToken._MAX` are never changed after assignment, so they can be declared as `constant`. A big advantage of constant variables is that reading them is significantly cheaper than reading from regular state variables.

Recommendation

We recommend using constant state variables for `FabweltToken._GRANULARITY` and `FabweltToken._MAX` in order to save gas.

TCK-04 | Lack of Event Emissions for Significant Transactions

Category	Severity	Location	Status
Coding Style	Informational	projects/fabwelt/contracts/fabweitToken.sol (cd81a41): 572, 598, 607, 620, 624	Pending

Description

The following functions update crucial state variables. Events should be emitted to log these updates.

- `FabweltToken.deliver()`
- `FabweltToken.excludeAccount()`
- `FabweltToken.includeAccount()`
- `FabweltToken.setAsCharityAccount()`
- `FabweltToken.updateFee()`

Recommendation

We advise adding events for sensitive actions in the aforementioned functions and emitting them in the corresponding functions.

TCK-05 | Redundant Code

Category	Severity	Location	Status
Gas Optimization, Coding Style	Informational	projects/fabwelt/contracts/fabweitToken.sol (cd81a41): 653, 664~665, 745, 832	Pending

Description

In the function `FabweltToken._transfer()`, `takeFee` is set to `false` if `recipient` is excluded:

```
649         if (FeeAddress == sender || FeeAddress == recipient ||
_isExcluded[recipient]) {
650             takeFee = false;
651         }
652
653         if (StakeAddress == sender || StakeAddress == recipient ||
_isExcluded[recipient]) {
654             takeFee = false;
655         }
```

Checking `_isExcluded[recipient]` in the second `if` statement is unnecessary because it has been checked in the first `if` statement.

In the function `FabweltToken._transfer()`, the condition `!_isExcluded[sender] && !_isExcluded[recipient]` is unnecessary because `FabweltToken._transferStandard()` will be executed in the `else` statement:

```
664         ... else if (!_isExcluded[sender] && !_isExcluded[recipient]) {
665             _transferStandard(sender, recipient, amount);
666         } ...
```

The function `FabweltToken._reflectFee()` contains the following line:

```
745         _tTotal = _tTotal;
```

However, this does not bring any changes.

The function `FabweltToken._getTaxFee()` is a private function but never used within the contract.

Recommendation

We recommend removing the aforementioned redundant code or adding relevant logic if they are designed for some purpose.

TCK-06 | Unused Local Variable

Category	Severity	Location	Status
Gas Optimization	Informational	projects/fabwelt/contracts/fabweitToken.sol (cd81a41): 676, 691, 708, 724	Pending

Description

In the contract `FabweltToken`, the functions `_transferStandard()`, `_transferToExcluded()`, `_transferFromExcluded()`, and `_transferBothExcluded()` all contain the following line:

```
        uint256 currentRate =  _getRate();
```

However, the local variable `currentRate` is unused in the above mentioned functions.

Recommendation

We recommend implementing a use case or removing the variable.

TCK-07 | Check Allowance Before Transfer

Category	Severity	Location	Status
Gas Optimization	Informational	projects/fabwelt/contracts/fabweitToken.sol (cd81a41): 541~542	Pending

Description

In the function `FabweltToken.transferFrom()`, the function transfers tokens before checking if the message sender has enough of an allowance.

```
541         _transfer(sender, recipient, amount);
542         _approve(sender, _msgSender(), _allowances[sender][_msgSender()].sub(amount,
"TOKEN20: transfer amount exceeds allowance"));
```

It would be better to validate the allowance first to reduce gas costs in case it reverts.

Recommendation

We recommend checking there is sufficient allowance before initiating a transfer.

TCK-08 | Missing Error Messages

Category	Severity	Location	Status
Coding Style	Informational	projects/fabwelt/contracts/fabweitToken.sol (cd81a41): 625	Pending

Description

The **require** statement can be used to check for conditions and throw an exception if the condition is not met. It is better to provide a string message containing details about the error that will be passed back to the caller.

Recommendation

We advise refactoring the linked codes as below:

```
625         require(_txFee < 100 && _stakeFee < 100 && _charityFee < 100, "Invalid fee
rates");
```

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.

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.

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.