Security Assessment

French Connection Finance

Dec 12th, 2021

Table of Contents

Summary

This report has been prepared for French Connection Finance to discover issues and vulnerabilities in the source code of the French Connection Finance 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	French Connection Finance
Platform	BSC
Language	Solidity
Codebase	https://bscscan.com/token/0x4673f018cc6d401aad0402bdbf2abcbf43dd69f3
Commit	

Audit Summary

Delivery Date	Dec 12, 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	3	0	0	0	0	3
Medium	0	0	0	0	0	0
Minor	3	0	0	3	0	0
Informational	7	0	0	7	0	0
Discussion	0	0	0	0	0	0

Audit Scope

ID	File	SHA256 Checksum
FCC	FCF.sol	955df27eaa021466fdd1f6b7fb5555391cbc9ad0fd032062bb2e413d970ce2dd

Findings

Critical	**0** (0.00%)
Major	**3** (23.08%)
Medium	**0** (0.00%)
Minor	**3** (23.08%)
Informational	**7** (53.85%)
Discussion	**0** (0.00%)

ID	Title	Category	Severity	Status
FCC-01	Unsafe Cast	Logical Issue	Minor	Acknowledged
FCC-02	Third Party Dependencies	Volatile Code	Minor	Acknowledged
FCC-03	Centralization Risk	**Centralization / Privilege**	**Major**	Resolved
FCC-04	Missing Emit Events	Coding Style	Informational	Acknowledged
FCC-05	Centralized Risk in `addLiquidity()`	**Centralization / Privilege**	**Major**	Resolved
FCC-06	Return Value Not Handled	Volatile Code	Informational	Acknowledged
FCC-07	Potential Sandwich Attack	Logical Issue	Informational	Acknowledged
FCC-08	Variable Name Typo	Coding Style	Informational	Acknowledged
FCC-09	Missing Input Validation	Volatile Code	Minor	Acknowledged
FCC-10	Unlocked Compiler Version	Language Specific	Informational	Acknowledged
FCC-11	Missing Error Messages	Coding Style	Informational	Acknowledged
FCC-12	Centralization Risk with `safeManager` Role	**Centralization / Privilege**	**Major**	Resolved
FCC-13	Ambiguous Implementation	Coding Style	Informational	Acknowledged

FCC-01 | Unsafe Cast

Category	Severity	Location	Status
Logical Issue	Minor	projects/FrenchConnectionFinance/FCF.sol (e94865b): 1231~1236, 1587, 1592	Acknowledged

Description

Here is the original description text of an old finding from that project that may apply to the current project.

The linked statements cast a `uint256` value to an `int256` without evaluating its bounds.

Recommendation

We advise a safe casting operation to be performed by ensuring the result is still positive as high numbers will cause an underflow to occur here, thereby causing the system to misbehave.

Alleviation

`[French Connection Finance team]`: Acknowledged but can't modify it

FCC-02 | Third Party Dependencies

Category	Severity	Location	Status
Volatile Code	Minor	projects/FrenchConnectionFinance/FCF.sol (e94865b): 1913	Acknowledged

Description

Here is the original description text of an old finding from that project that may apply to the current project.

The contract is serving as the underlying entity to interact with third-party protocols, including:

- `IUniswapV2Router02`

The scope of the audit treats 3rd party entities as black boxes and assumes their functional correctness. However, in the real world, 3rd parties can be compromised, which may lead to lost or stolen assets. In addition, upgrades of 3rd parties can create severe impacts, such as increasing fees of 3rd parties, migrating to new LP pools, etc.

Recommendation

We understand that the business logic of FCF requires interaction with the aforementioned protocols. We encourage the team to constantly monitor the status of 3rd parties to mitigate side effects when unexpected activities are observed.

Alleviation

`[French Connection Finance team]`: Acknowledged but can't modify it

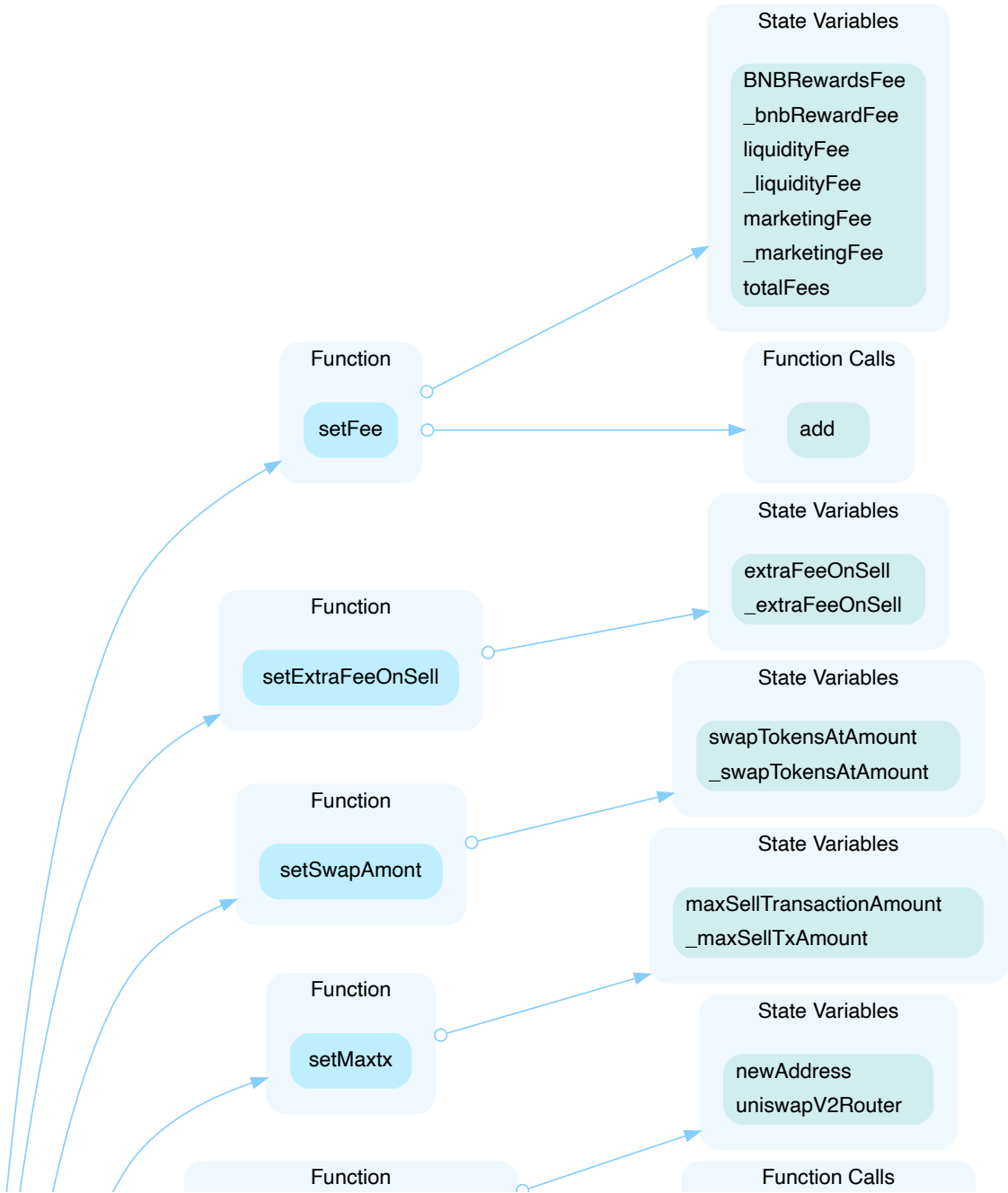
FCC-03 | Centralization Risk

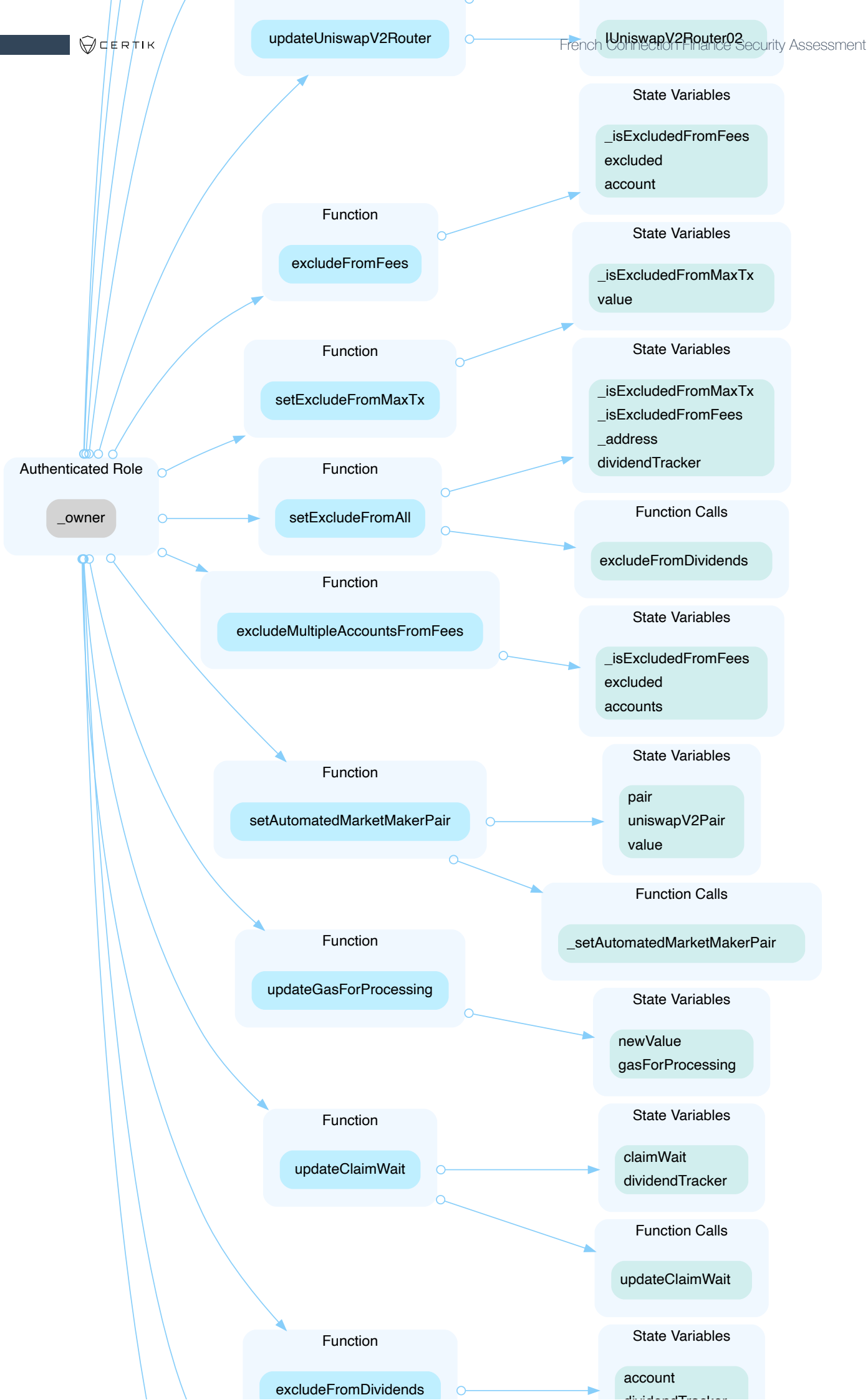
Category	Severity	Location	Status
Centralization / Privilege	Major	projects/FrenchConnectionFinance/FCF.sol (e94865b): 1840~1846, 1848~1850, 1852~1854, 1856~1858, 1910~1914, 1916~1921, 1923~1925, 1927~1931, 1933~1939, 1941~1945, 1958~1963, 1965~1967, 2037~2039, 2041~2044, 2046~2048, 1548~1556, 1558~1563, 1647~1662, 1711~1721, 1759~1761, 1763~1767	Resolved

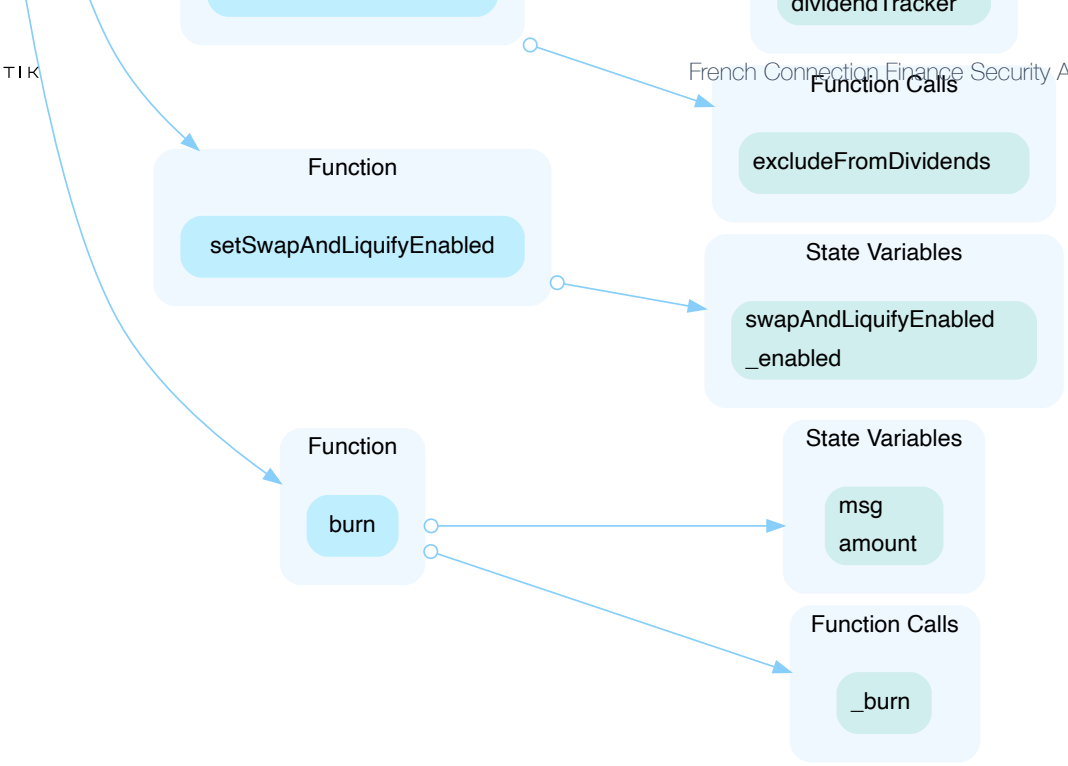
Description

In the contract, `FCF`, the role, `_owner`, has the authority over the functions shown in the diagram below.

Any compromise to the privileged account which has access to `_owner` may allow the hacker to take advantage of this.

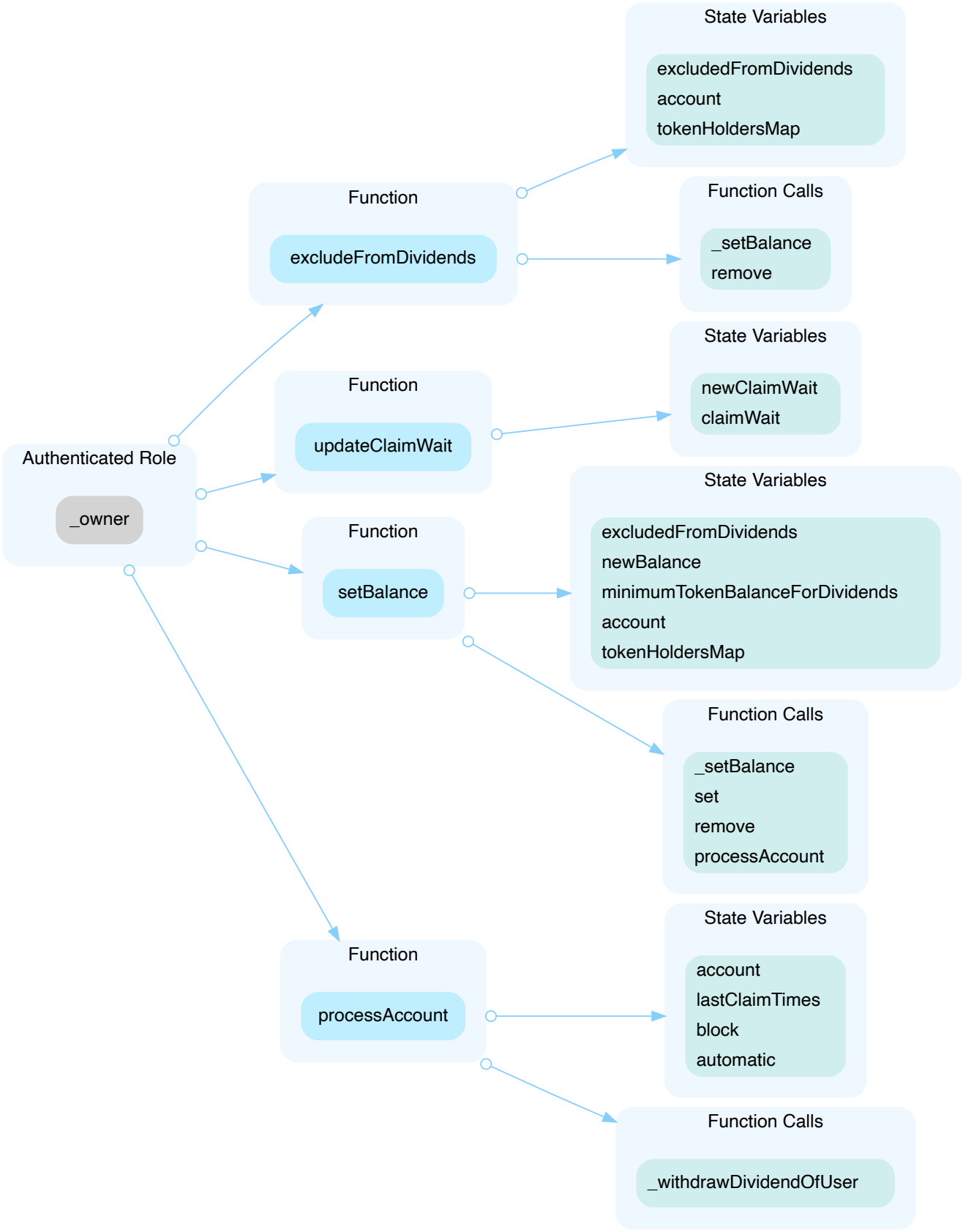



updateUniswapV2Router
IUniswapV2Router02
State Variables
_isExcludedFromFees
excluded
account
Function
excludeFromFees
State Variables
_isExcludedFromMaxTx
value
Function
setExcludeFromMaxTx
State Variables
_isExcludedFromMaxTx
_isExcludedFromFees
_address
dividendTracker
Authenticated Role
_owner
Function
setExcludeFromAll
Function Calls
excludeFromDividends
Function
excludeMultipleAccountsFromFees
State Variables
_isExcludedFromFees
excluded
accounts
Function
setAutomatedMarketMakerPair
State Variables
pair
uniswapV2Pair
value
Function Calls
_setAutomatedMarketMakerPair
Function
updateGasForProcessing
State Variables
newValue
gasForProcessing
Function
updateClaimWait
State Variables
claimWait
dividendTracker
Function Calls
updateClaimWait
Function
excludeFromDividends
State Variables
account



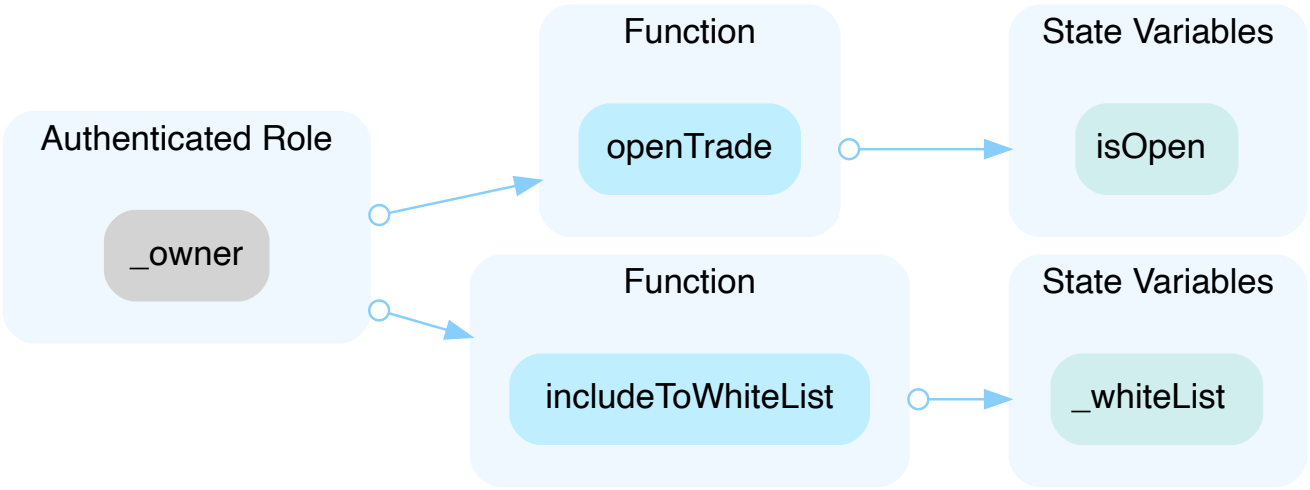
In the contract, `FCFDividendTracker`, the role, `_owner`, has the authority over the functions shown in the diagram below.

Any compromise to the privileged account which has access to `_owner` may allow the hacker to take advantage of this.



In the contract, `LockToken`, the role, `_owner`, has the authority over the functions shown in the diagram below.

Any compromise to the privileged account which has access to `_owner` may allow the hacker to take advantage of this.



Recommendation

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 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

`[French Connection Finance team]`: The client implemented the multiple signer control in the FCF.sol via the gnosis safe as recommended. The gnosis safe contract wallet has been deployed with address 0xFb3dF01C78DdFFce4BfEB6bE3963487B09d9E8C3 and transaction hash 0x81f3fd65ee375a3b57f265f9af23981e41d10c849ca04622a3121183ee663c85.

The FCF.sol contract has been deployed with address 0x4673f018cc6d401aad0402bdbf2abcbf43dd69f3 and transaction hash 0x68cdc73231e248c79eefb21aecd5d9c30cd4c538f3474410f063b7d98c76212c.

The owner of the FCF.sol contract is the address of the gnosis safe contract wallet 0xa6b71e26c5e0845f74c812102ca7114b6a896ab2 which has been configured with a 3 out of 3 signature threshold. With this development the FCF.sol has now multiple owners:

- The Ceo (FrenchConnected) John Nasr: 0xCD0664a86B587f671b4af36fa99112b676a6c012
- COO DJfrenchfellas: 0xe7b90650FF8B506AD5BcA70baD2B35a5353E08ff
- Community member: 0x5bfEa50e3B1bA3A585b73F72C1882a477591C830

The details of the gnosis safe deployment can be found in https://www.reddit.com/r/Frenchconnectiontoken/comments/ralpok/multi_signature_wallet_certik_audit_requirement/

FCC-04 | Missing Emit Events

Category	Severity	Location	Status
Coding Style	Informational	projects/FrenchConnectionFinance/FCF.sol (e94865b): 1647~1662, 1731~1733, 1735~1738, 1740~1743, 1759~1761, 1763~1767, 1840~1846, 1848~1850, 1852~1854, 1856~1858, 1923~1925, 1927~1931, 1941~1945, 1965~1967, 2037~2039, 2046~2048	Acknowledged

Description

There should always be events emitted in the sensitive functions listed above that are controlled by centralization roles or can modify the sensitive variables of the project.

Recommendation

It is recommended emitting events for the sensitive functions that are controlled by centralization roles or can affect sensitive functionalities/variables of the project.

Alleviation

`[French Connection Finance team]`: Acknowledged but can't modify it

FCC-05 | Centralized Risk in `addLiquidity()`

Category	Severity	Location	Status
Centralization / Privilege	Major	projects/FrenchConnectionFinance/FCF.sol (e94865b): 2186	Resolved

Description

```
2180 // add the liquidity
2181 uniswapV2Router.addLiquidityETH{value: ethAmount}(
2182     address(this),
2183     tokenAmount,
2184     0, // slippage is unavoidable
2185     0, // slippage is unavoidable
2186     owner(),
2187     block.timestamp
2188 );
```

The `addLiquidity` function calls the `uniswapV2Router.addLiquidityETH` function with the `to` address specified as `owner()` for acquiring the generated LP tokens from the `FCF-BNB` pool. As a result, over time the `_owner` address will accumulate a significant portion of LP tokens.If the `_owner` is an EOA (Externally Owned Account), mishandling of its private key can have devastating consequences to the project as a whole.

Recommendation

We advise the `to` address of the `uniswapV2Router.addLiquidityETH` function call to be replaced by the contract itself, i.e. `address(this)`, and to restrict the management of the LP tokens within the scope of the contract's business logic. This will also protect the LP tokens from being stolen if the `_owner` account is compromised. In general, we strongly recommend centralized privileges or roles in the protocol to be improved via a decentralized mechanism or via smart-contract based accounts with enhanced security practices, f.e. Multisignature wallets.

Indicatively, here are some feasible solutions that would also mitigate the potential risk:

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

Alleviation

`[French Connection Finance team]`: The LP are periodically sent to time locks on DX sale once (LP token are going to be sent to the multisig)

`[French Connection Finance team]`: The client implemented the multiple signer control in the FCF.sol via the gnosis safe as recommended. The gnosis safe contract wallet has been deployed with address 0xFb3dF01C78DdFFce4BfEB6bE3963487B09d9E8C3 and transaction hash 0x81f3fd65ee375a3b57f265f9af23981e41d10c849ca04622a3121183ee663c85.

The FCF.sol contract has been deployed with address 0x4673f018cc6d401aad0402bdbf2abcbf43dd69f3 and transaction hash 0x68cdc73231e248c79eefb21aecd5d9c30cd4c538f3474410f063b7d98c76212c.

The owner of the FCF.sol contract is the address of the gnosis safe contract wallet 0xa6b71e26c5e0845f74c812102ca7114b6a896ab2 which has been configured with a 3 out of 3 signature threshold. With this development the FCF.sol has now multiple owners:

- The Ceo (FrenchConnected) John Nasr: 0xCD0664a86B587f671b4af36fa99112b676a6c012
- COO DJfrenchfellas: 0xe7b90650FF8B506AD5BcA70baD2B35a5353E08ff
- Community member: 0x5bfEa50e3B1bA3A585b73F72C1882a477591C830

The details of the gnosis safe deployment can be found in https://www.reddit.com/r/Frenchconnectiontoken/comments/ralpok/multi_signature_wallet_certik_audit_requirement/

FCC-06 | Return Value Not Handled

Category	Severity	Location	Status
Volatile Code	Informational	projects/FrenchConnectionFinance/FCF.sol (e94865b): 2181	Acknowledged

Description

The return values of function `addLiquidityETH` is not properly handled.

```
2181 uniswapV2Router.addLiquidityETH{value: ethAmount}(
2182     address(this),
2183     tokenAmount,
2184     0, // slippage is unavoidable
2185     0, // slippage is unavoidable
2186     owner(),
2187     block.timestamp
2188 );
```

Recommendation

We recommend using variables to receive the return value of the functions mentioned above and handle both success and failure cases if needed by the business logic.

Alleviation

`[French Connection Finance team]`: Acknowledged but can't modify it

FCC-07 | Potential Sandwich Attack

Category	Severity	Location	Status
Logical Issue	Informational	projects/FrenchConnectionFinance/FCF.sol (e94865b): 2164, 2181	Acknowledged

Description

Potential sandwich attacks could happen if calling `uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens` and `uniswapV2Router.addLiquidityETH` without setting restrictions on slippage.

For example, when we want to make a transaction of swapping 100 Token A for 1 ETH, an attacker could raise the price of ETH by adding Token A into the pool before the transaction so we might only get 0.1 ETH. After the transaction, the attacker would be able to withdraw more than he deposited because the total value of the pool increases by 0.9 ETH.

Recommendation

We recommend using Oracle to get an estimation of prices and setting minimum amounts based on the prices when calling the aforementioned functions.

Alleviation

`[French Connection Finance team]`: Acknowledged but can't modify it

FCC-08 | Variable Name Typo

Category	Severity	Location	Status
Coding Style	Informational	projects/FrenchConnectionFinance/FCF.sol (e94865b): 1816	Acknowledged

Description

In the following code snippet, `tokensIntoLiqudity` should be `tokensIntoLiquidity`.

```
event SwapAndLiquify(
    uint256 tokensIntoLiqudity,
    uint256 ethReceived
);
```

Recommendation

We recommend correcting all typos in the contract.

Alleviation

`[French Connection Finance team]`: Acknowledged

FCC-09 | Missing Input Validation

Category	Severity	Location	Status
Volatile Code	Minor	projects/FrenchConnectionFinance/FCF.sol (e94865b): 1910, 1941	Acknowledged

Description

The variables `newAddress`, `pair` should be verified as non-zero values to prevent being mistakenly assigned as `address(0)` in the `updateUniswapV2Router()`, and `setAutomatedMarketMakerPair()` functions respectively.

Recommendation

We advise the client to check that the addresses are not zero in `updateUniswapV2Router()`, `setAutomatedMarketMakerPair()` like as follows:

```
require(newAddress != address(0), "FCF: newAddress is a zero address");
```

Alleviation

`[French Connection Finance team]`: Acknowledged, Multisig will prevent setting to 0

FCC-10 | Unlocked Compiler Version

Category	Severity	Location	Status
Language Specific	Informational	projects/FrenchConnectionFinance/FCF.sol (e94865b)	Acknowledged

Description

The contract contains unlocked compiler versions. An unlocked compiler version in the contract's source code 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 difficult to identify over a span of multiple compiler versions rather than a specific one.

Recommendation

It is general practice to alternatively lock the compiler at a specific version rather than allow a range of compiler versions to be utilized to avoid compiler-specific bugs and thus be able to detect emerging ones. We recommend locking the compiler at the lowest possible version that supports all the capabilities required by the codebase. This will ensure that the project utilizes a compiler version that has been in use for the longest time and as such is less likely to contain yet-undiscovered bugs.

Alleviation

`[French Connection Finance team]`: Acknowledged but can't modify it

FCC-11 | Missing Error Messages

Category	Severity	Location	Status
Coding Style	Informational	projects/FrenchConnectionFinance/FCF.sol (e94865b): 1735, 1740, 1548, 1396	Acknowledged

Description

The convenience function `require` can be used to check for conditions and throw an exception if the condition is not met. If you do not provide a string argument to require, it will revert with empty error data, not even including the error selector. For example:

- `distributeDividends()`
- `excludeFromDividends()`
- `withdraw()`
- `withdrawBNB()`

Reference: https://docs.soliditylang.org/en/v0.8.4/control-structures.html?highlight=require#panic-via-assert-and-error-via-require

Recommendation

We advise the client to add error messages.

Alleviation

`[French Connection Finance team]`: Acknowledged but can't modify it

FCC-12 | Centralization Risk with `safeManager` Role

Category	Severity	Location	Status
Centralization / Privilege	**Major**	projects/FrenchConnectionFinance/FCF.sol (e94865b): 1735, 1740	Resolved

Description

In the contract `FCF`, the role `safeManager` has the authority over the following function:

- `withdraw()`
- `withdrawBNB()`

Any compromise to the `safeManager` account may allow the hacker to take advantage of this and transfer any amount of BNB/specific token to the address of `safeManager`.

Recommendation

We advise the client to carefully manage the `safeManager` 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

`[French Connection Finance team]`: The client implemented the multiple signer control in the FCF.sol via the gnosis safe as recommended. The gnosis safe contract wallet has been deployed with address 0xFb3dF01C78DdFFce4BfEB6bE3963487B09d9E8C3 and transaction hash 0x81f3fd65ee375a3b57f265f9af23981e41d10c849ca04622a3121183ee663c85.

The FCF.sol contract has been deployed with address 0x4673f018cc6d401aad0402bdbf2abcbf43dd69f3 and transaction hash 0x68cdc73231e248c79eefb21aecd5d9c30cd4c538f3474410f063b7d98c76212c.

The owner of the FCF.sol contract is the address of the gnosis safe contract wallet 0xa6b71e26c5e0845f74c812102ca7114b6a896ab2 which has been configured with a 3 out of 3 signature threshold. With this development the FCF.sol has now multiple owners:

- The Ceo (FrenchConnected) John Nasr: 0xCD0664a86B587f671b4af36fa99112b676a6c012
- COO DJfrenchfellas: 0xe7b90650FF8B506AD5BcA70baD2B35a5353E08ff
- Community member: 0x5bfEa50e3B1bA3A585b73F72C1882a477591C830

The details of the gnosis safe deployment can be found in https://www.reddit.com/r/Frenchconnectiontoken/comments/ralpok/multi_signature_wallet_certik_audit_requirement/

FCC-13 | Ambiguous Implementation

Category	Severity	Location	Status
Coding Style	Informational	projects/FrenchConnectionFinance/FCF.sol (e94865b): 1916	Acknowledged

Description

Based on the suggestion of the name, the function `excludeFromFees()` is supposed to exclude a certain address from the fee. However, in the current implementation, this function can either be used to exclude an included address or be used to include an excluded address, which does not align with the suggestion of the name and will cause confusion to the user.

Recommendation

WE advise the client to consider implementing the function properly and/or add more specific documentation to the function.

Alleviation

`[French Connection Finance team]`: Acknowledged but can't modify it

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.

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.

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.