Code Security Assessment

Eigen Network

Mar 4th, 2022

Table of Contents

Summary

This report has been prepared for Eigen Network to discover issues and vulnerabilities in the source code of the Eigen Network 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	Eigen Network
Platform	Ethereum
Language	Solidity
Codebase	https://github.com/ieigen/NCW
Commit	83a776ffc973f8e079289b699df5ed95f450c6b9

Audit Summary

Delivery Date	Mar 04, 2022 UTC
Audit Methodology	Static Analysis, Manual Review

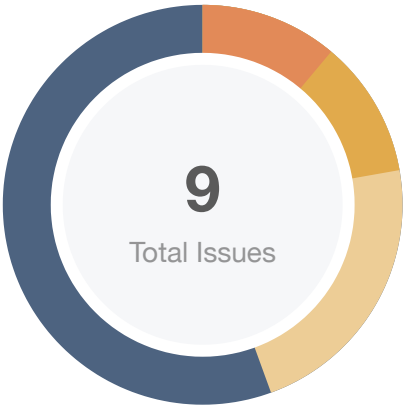
Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Partially Resolved	Mitigated	Resolved
Critical	0	0	0	0	0	0	0
Major	1	0	0	0	0	0	1
Medium	1	0	0	0	0	1	0
Minor	2	0	0	0	0	0	2
Informational	5	0	0	1	0	0	4
Discussion	0	0	0	0	0	0	0

Audit Scope

ID	File	SHA256 Checksum
BME	BaseModule.sol	332f7ad29e42f4c40d28d73c4fd929f4b1a610724f2274e10de398b10caa6ae9
FEN	Forwarder.sol	d4e50b985c452d29b60229bed6d3e119647a4ec3051c4ddadc400a89b3f28cb0
FTE	ForwarderTarget.sol	5a3777d06fb668b64cff456807f0b60f29ef6080b68a0a26f1e6b87391c516bc
MRE	ModuleRegistry.sol	9029226b615e56bd46cc83674e51ccedbabd56b3a2af3712cafa1c113d7f32ed
SME	SecurityModule.sol	a04f2d21468a6b67179d76af5f1e7d5e270c2034e16fcb247fa20cc735457441
TTE	TestToken.sol	5506fd53ccf28a1018c6d2c2e36ed3f23fae2ebe80e7e4a88e094f907a36ef36
TME	TransactionModule.sol	cbe45b394a874851a96e201a286a83cb891bb44ece17207701e18d57fa13a7dc
WEN	Wallet.sol	4d8ac7307570a9cdcb5cefc37ba05f81449d59583aff82015173ed5040b2f518

Findings



Critical	**0** (0.00%)
Major	**1** (11.11%)
Medium	**1** (11.11%)
Minor	**2** (22.22%)
Informational	**5** (55.56%)
Discussion	**0** (0.00%)

ID	Title	Category	Severity	Status
Eigen-01	Centralization Risks	**Centralization / Privilege**	**Medium**	Mitigated
Eigen-02	Financial Models	Logical Issue	Informational	Resolved
BME-01	Missing Duplicate Check	Logical Issue	Minor	Resolved
ENC-01	Missing Emit Events	Coding Style	Informational	Resolved
ENC-02	Assembly Usage	Language Specific	Informational	Acknowledged
FEN-01	Potential Out-of-Gas Exception	Gas Optimization	Informational	Resolved
FEN-02	Missing Error Messages	Coding Style	Informational	Resolved
SME-01	Function never worked	Logical Issue	Major	Resolved
WEN-01	Authorization Issue	Logical Issue	Minor	Resolved

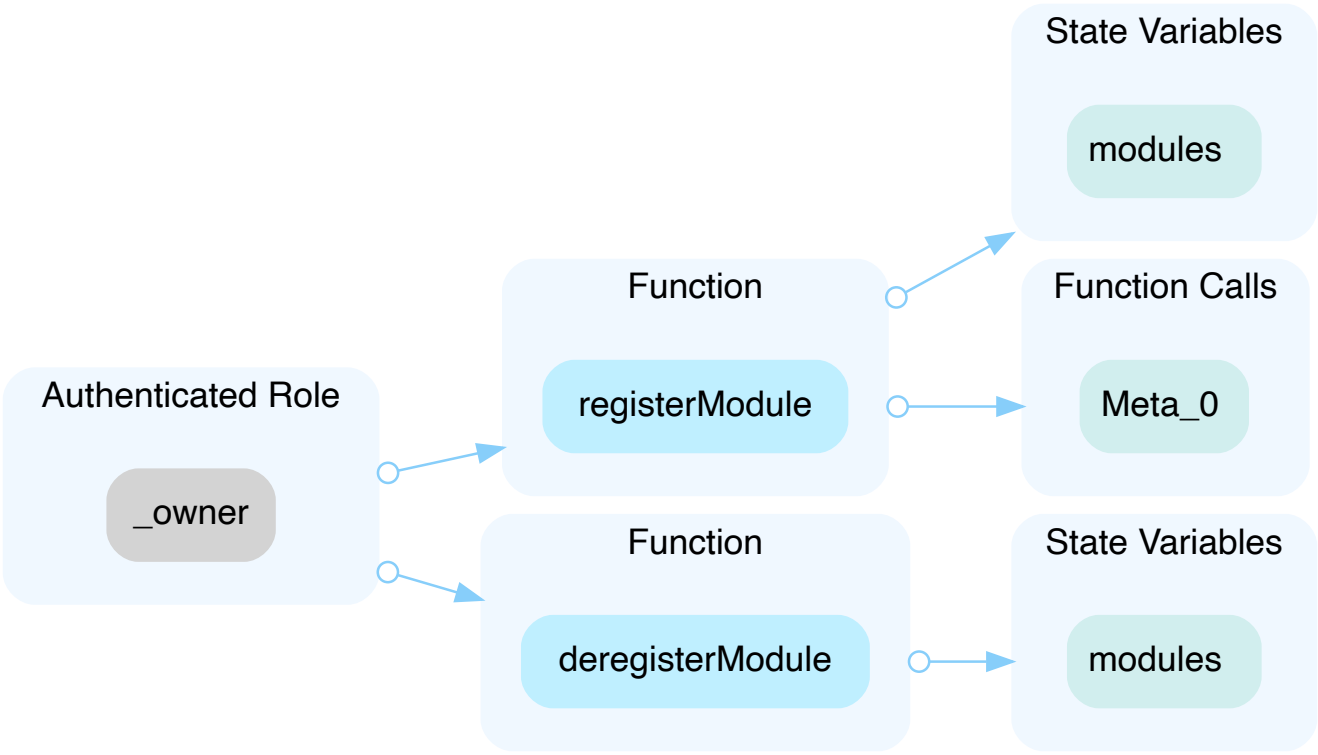
Eigen-01 | Centralization Risks

Category	Severity	Location	Status
Centralization / Privilege	Medium	Global	Mitigated

Description

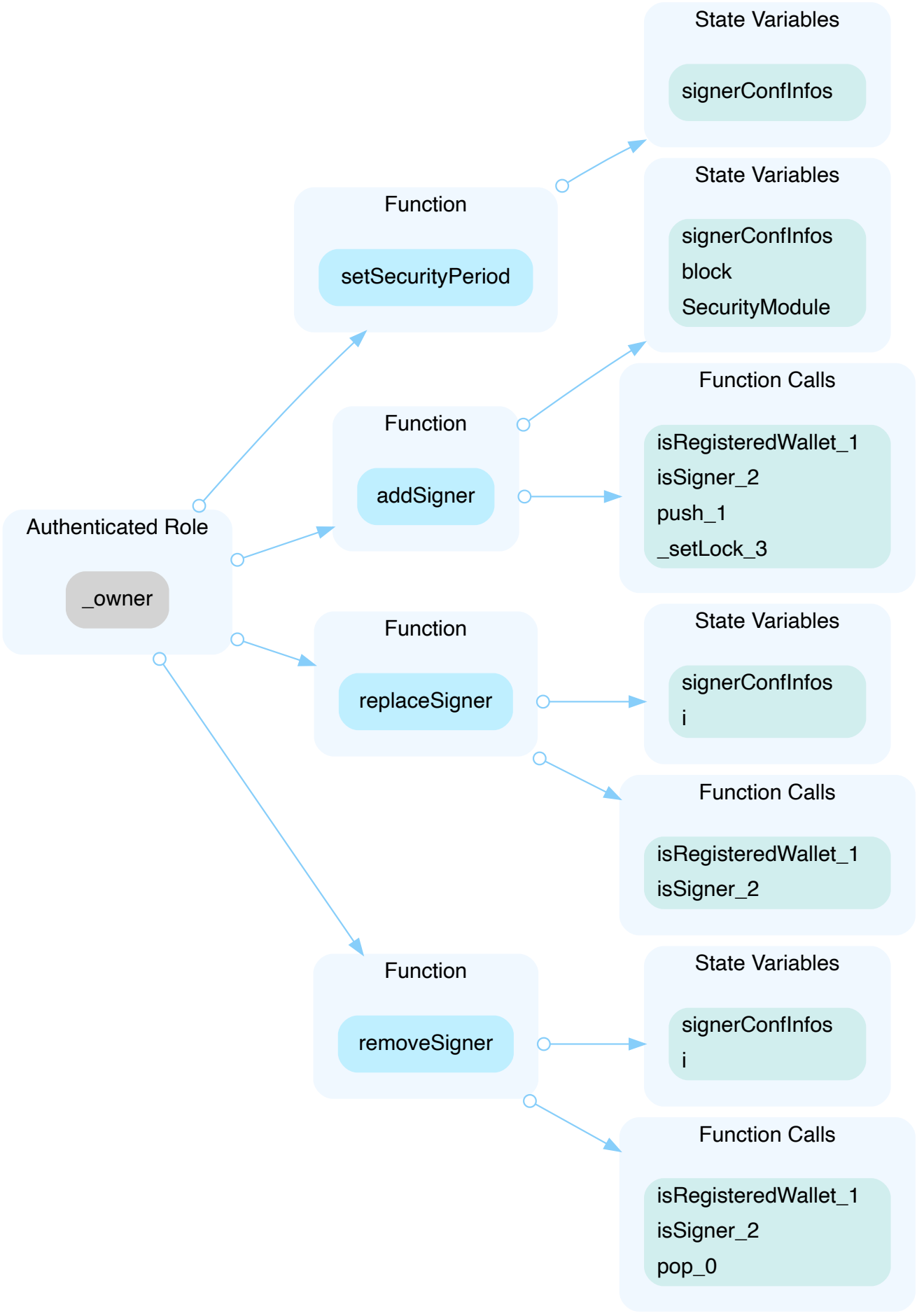
In the contract `ModuleRegistry` the role `_owner` has authority over the functions shown in the diagram below.

Any compromise to the `_owner` account may allow the hacker to take advantage of this authority.



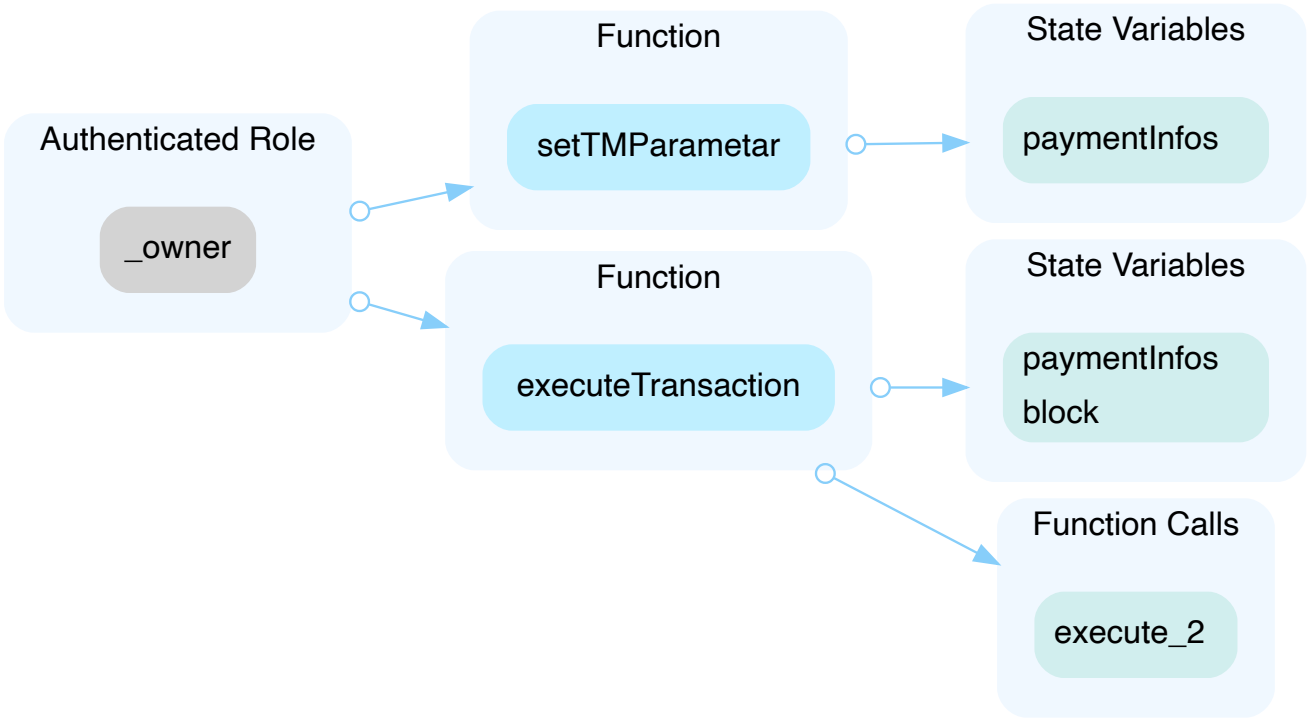
In the contract `SecurityModule` the role `_owner` has authority over the functions shown in the diagram below.

Any compromise to the `_owner` account may allow the hacker to take advantage of this authority.


State Variables
signerConfInfos
Function
setSecurityPeriod
State Variables
signerConfInfos
block
SecurityModule
Function
addSigner
Function Calls
isRegisteredWallet_1
isSigner_2
push_1
_setLock_3
Authenticated Role
_owner
Function
replaceSigner
State Variables
signerConfInfos
i
Function Calls
isRegisteredWallet_1
isSigner_2
Function
removeSigner
State Variables
signerConfInfos
i
Function Calls
isRegisteredWallet_1
isSigner_2
pop_0

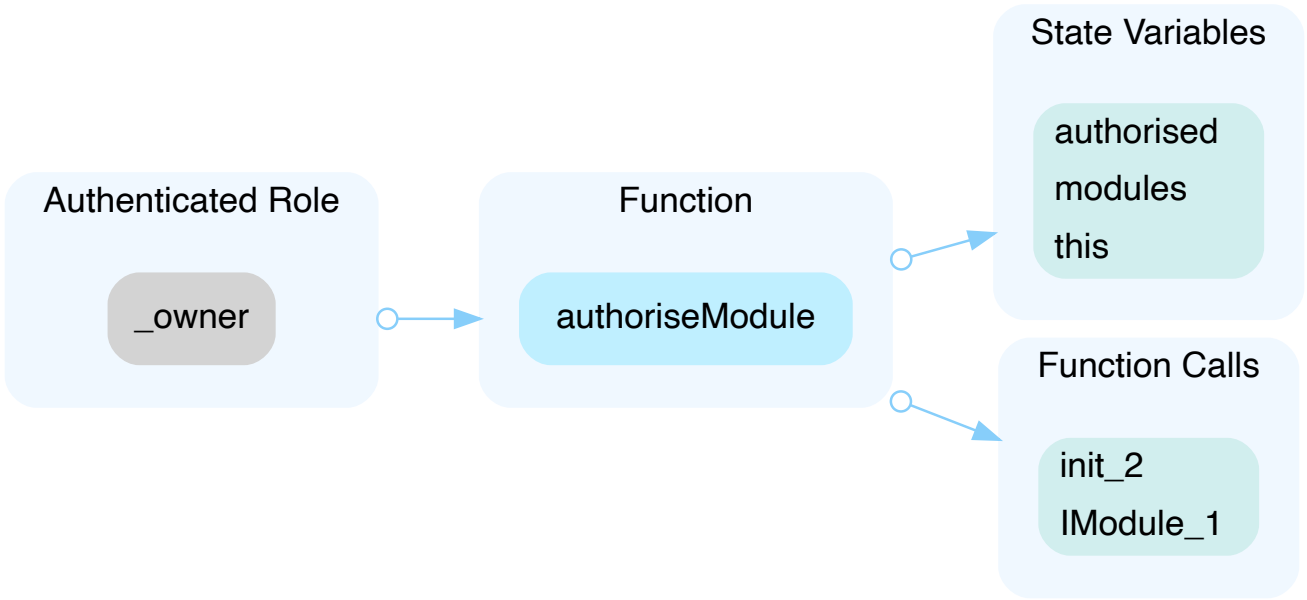
In the contract `TransactionModule` the role `_owner` has authority over the functions shown in the diagram below.

Any compromise to the `_owner` account may allow the hacker to take advantage of this authority.



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

Any compromise to the `_owner` account may allow the hacker to take advantage of this authority.



Recommendation

The risk describes the current project design and potentially makes iterations to improve in the security operation and level of decentralization, which in most cases cannot be resolved entirely at the present stage. 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 be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., multisignature wallets.

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

Short Term:

Timelock and Multi sign (⅔, ⅗) combination *mitigate* by delaying the sensitive operation and avoiding a single point of key management failure.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key compromised;
  AND
- A medium/blog link for sharing the timelock contract and multi-signers addresses information with the public audience.

Long Term:

Timelock and DAO, the combination, *mitigate* by applying decentralization and transparency.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Introduction of a DAO/governance/voting module to increase transparency and user involvement.
  AND
- A medium/blog link for sharing the timelock contract, multi-signers addresses, and DAO information with the public audience.

Permanent:

Renouncing the ownership or removing the function can be considered *fully resolved*.

- Renounce the ownership and never claim back the privileged roles.
  OR
- Remove the risky functionality.

Alleviation

[Eigen]: Firstly we define our multi-sig wallet as a personal security-enhancement wallet. And for the user's key, we provide a self-custodian mechanism to improve its safety.

Secondly, for some high-risk operations, we secure them by multi-sig and calm-down lock as CertiK's advise in the report.

For ModuleRegistry, this is not an issue, because the user's asset is safe and can be redeemed no matter what happened to the common module. We'll import a DAO instead of we ourselves to maintain the development, upgrade, and bug fix for the common module in the future.

For SecurityModule and TransactionModule, we adopt multi-sig to secure set* methods. Every operation that modifies wallet-related parameters needs to use multi-signature to reduce the risk of centralization. Meanwhile, multi-signature is used to secure `addModule` method. In addition to the multi-signature mechanism, we also introduced locks to improve the security of the contract wallet. When users perform different operations, the wallet will add different locks to prevent users from frequent operations. Locks are divided into the global lock, signer-related lock, and transaction-related lock. Signer-related and transaction-related locks are local locks. To make an analogy, the global lock is like a door in a house, and the others are locks for small doors in the house.

For the wallet, only the registered module can call `authoriseModule`. And as mentioned above, there will be a DAO to manage the module registry in the future.

To sum up, we use the multi-signature and locking mechanism for more operations on the wallet to reduce centralization risks.

Eigen-02 | Financial Models

Category	Severity	Location	Status
Logical Issue	Informational	Global	Resolved

Description

This protocol provides the wallet which allows the user:

- Use multi-signature
- Recover the owner by social recovery
- Lockable
- Payment Limitation

But there are some issues needed to discuss:

1. What is the contract `Forwarder` used for?
2. What is the `_sequenceId`? what is the `_sequenceId` used for?
3. The comments say that transactions over the limit require multiple signatures, but we don't see how this is accomplished in the code.

Recommendation

Financial models of blockchain protocols need to be resilient to attacks. They need to pass simulations and verifications to guarantee the security of the overall protocol.

The financial model of this protocol is not in the scope of this audit.

Alleviation

The Eigen team removed the useless contract `Forwarder`. And the `_sequenceId` is used as nonce to order the transaction, and stop too old or new transaction from potential attack.

BME-01 | Missing Duplicate Check

Category	Severity	Location	Status
Logical Issue	Minor	BaseModule.sol: 98~101	Resolved

Description

The function did not check whether the `wallets` array had already contained the `_wallet`. However, it can be inferred from `removeWallet()` and `isRegisteredWallet()` that every element in the `wallets` array shouldn't have a duplicate.

Recommendation

Add duplicate check.

Alleviation

The Eigen team heeded our advice and added a require check for function `addWallet()`, the change was supplied in commit `f398176e1464204de29f50dba89a04a961f0a9ea`.

ENC-01 | Missing Emit Events

Category	Severity	Location	Status
Coding Style	Informational	TransactionModule.sol: 26~36, 38~48, 67~70, 72~84, 86~99 Wallet.sol: 96~99 SecurityModule.sol: 41~57, 64~67, 69~79, 131~141, 143~158, 160~177, 189~205, 207~212 ModuleRegistry.sol: 20~24, 30~34	Resolved

Description

There should always be events emitted in the sensitive functions that are controlled by centralization roles.

Recommendation

It is recommended emitting events for the sensitive functions that are controlled by centralization roles.

Alleviation

The Eigen team added events in the sensitive functions that are controlled by centralization roles, the change was supplied in commit `f398176e1464204de29f50dba89a04a961f0a9ea`.

ENC-02 | Assembly Usage

Category	Severity	Location	Status
Language Specific	Informational	SecurityModule.sol: 309~313 Wallet.sol: 180~183, 195~198	Acknowledged

Description

File: projects/EigenNetwork/contracts/SecurityModule.sol (Line 309-313, Function `SecurityModule.splitSignature`)

```
assembly {
  r := mload(add(_signatures, add(0x20,mul(0x41,_index))))
  s := mload(add(_signatures, add(0x40,mul(0x41,_index))))
  v := and(mload(add(_signatures, add(0x41,mul(0x41,_index)))), 0xff)
}
```

File: projects/EigenNetwork/contracts/Wallet.sol (Line 180-183, Function `Wallet.invoke`)

```
assembly {
  returndatacopy(0, 0, returndatasize())
  revert(0, returndatasize())
}
```

File: projects/EigenNetwork/contracts/Wallet.sol (Line 195-198, Function `Wallet.raw_invoke`)

```
assembly {
  returndatacopy(0, 0, returndatasize())
  revert(0, returndatasize())
}
```

Recommendation

We advise against using EVM assembly, as it is error-prone.

Alleviation

The Eigen team acknowledged this finding.

FEN-01 | Potential Out-of-Gas Exception

Category	Severity	Location	Status
Gas Optimization	Informational	Forwarder.sol: 33~36	Resolved

Description

The fallback function can only rely on 2300 gas being available (for example when sending or transfer is used), token transfer in the fallback function will cost more than 2300 gas and more extra gas is required. Is that designed as expected? (https://docs.soliditylang.org/en/v0.5.8/contracts.html#fallback-function)

Recommendation

Reconsider the initial design and decide whether to change it.

Alleviation

The Eigen team removed the related code, the change was supplied in commit `168f15e63137a3ba63633e029707f6be2126c08d`.

FEN-02 | Missing Error Messages

Category	Severity	Location	Status
Coding Style	Informational	Forwarder.sol: 23~25	Resolved

Description

It will be confusing if the modifier just revert directly without providing error message.

Recommendation

We suggest the function use `require` instead of `revert` and provide the error message. Such as,

```
modifier onlyParent {
  require(msg.sender == parentAddress, "error message");
  _;
}
```

Alleviation

The Eigen team removed the related code, the change was supplied in commit `168f15e63137a3ba63633e029707f6be2126c08d`.

SME-01 | Function Never Worked

Category	Severity	Location	Status
Logical Issue	Major	SecurityModule.sol: 66	Resolved

Description

The `addModule` calls the `IWallet(_wallet).authoriseModule()`. The `IWallet(_wallet).authoriseModule()` needs the `owner` authority but the `SecurityModule` is not the owner. So the `authoriseModule()` will be reverted.

Recommendation

Check the design intention and fix it.

Alleviation

The Eigen team modified the function `addModule()` with the same authority check, which was provided in commit `587a5d4a3795410d12c44e8ddeb878f90ef7d8a6`.

WEN-01 | Authorization Issue

Category	Severity	Location	Status
Logical Issue	Minor	Wallet.sol: 81	Resolved

Description

If the owner uses the function `authoriseModule()` to cancel the authorization of the module, the module can never be authorized again. Because the `module.init()` need the `_wallet` was never registered but the authorization before had already registered the `_wallet`.

Recommendation

Review the original design and fix it.

Alleviation

The Eigen team heeded our advice and changed the function `authoriseModule()`, the change was supplied in commit `587a5d4a3795410d12c44e8ddeb878f90ef7d8a6`.

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.

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.