NUMEN

Smart Contract Audit Report

CoboSafe Smart Contract

16 Dec 2022

Numen Cyber Labs - Security Services

Table of Content

1 Executive Summary

Numen Cyber Technology was engaged by CoboSafe to review smart contract implementation. The assessment was conducted in accordance with our systematic approach to evaluate potential security issues based upon customer requirement. The report provides detailed recommendations to resolve the issue and provide additional suggestions or recommendations for improvement.

Two Medium severities findings are related to owner authority, centralized risk.

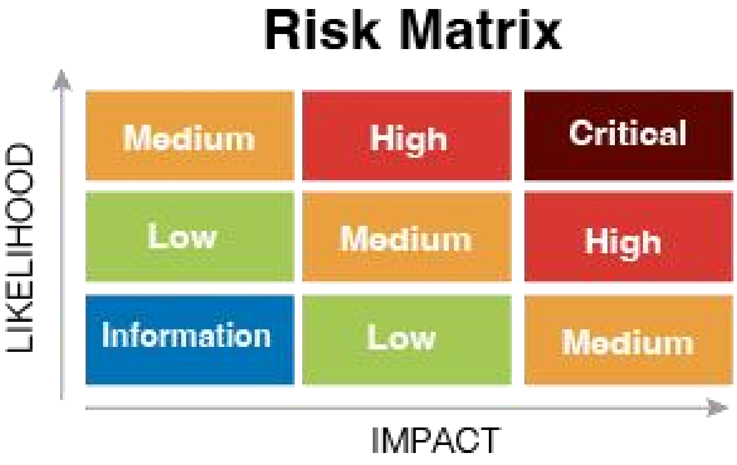
The outcome of the assessment outlined in chapter 3 provides the system's owners a full description of the vulnerabilities identified, the associated risk rating for each vulnerability, and detailed recommendations that will resolve the underlying technical issue.

Methodology

To standardize the evaluation, we define the following terminology based on OWASP Risk Rating Methodology [10] which is the gold standard in risk assessment using the following risk models:

- Likelihood: represents how likely a particular vulnerability is to be uncovered and exploited in the wild.
- Impact: measures the technical loss and business damage of a successful attack.
- Severity: determine the overall criticality of the risk.

Likelihood and impact are categorized into three ratings: High, Medium and Low. Severity is determined by likelihood and impact and can be classified into four categories accordingly, Critical, High, Medium, Low shown in table 1.1.


Risk Matrix
LIKELIHOOD
IMPACT

Medium	High	Critical
Low	Medium	High
Information	Low	Medium

Table 1.1: Overall Risk Severity

To evaluate the risk, we will be going through a list of items, and each would be labelled with a severity category. The audit was performed with a systematic approach guided by a comprehensive assessment list carefully designed to identify known and impactful security issues. If our tool or analysis does not identify any issue, the contract can be considered safe regarding the assessed item. For any discovered issue, we might further deploy contracts on our private test environment and run tests to confirm the findings. If necessary, we would additionally build a PoC to demonstrate the possibility of exploitation. The concrete list of check items is shown in Table 1.2.

- Basic Coding Bugs: We first statically analyze given smart contracts with our proprietary static code analyzer for known coding bugs, and then manually verify (reject or confirm) all the issues found by our tool.

- Code and business security testing: We further review business logics, examine system operations, and place DeFi-related aspects under scrutiny to uncover possible pitfalls and/or bugs.

- Additional Recommendations: We also provide additional suggestions regarding the coding and development of smart contracts from the perspective of proven programming practices.

Category	Assessment Item
Basic Coding Assessment	Apply Verification Control
	Authorization Access Control
	Forged Transfer Vulnerability
	Forged Transfer Notification
	Numeric Overflow
	Transaction Rollback Attack
	Transaction Block Stuffing Attack
	Soft fail Attack
	Hard fail Attack
	Abnormal Memo
	Abnormal Resource Consumption
	Secure Random Number
Advanced Source Code Scrutiny	Asset Security
	Cryptography Security
	Business Logic Review
	Source Code Functional Verification
	Account Authorization Control
	Sensitive Information Disclosure

	Circuit Breaker
	Blacklist Control
	System API Call Analysis
	Contract Deployment Consistency Check
Additional Recommendations	Semantic Consistency Checks
	Following Other Best Practices

Table 1.2: The Full List of Assessment Items

To better describe each issue we identified, we categorize the findings with Common Weakness Enumeration (CWE-699) [14], which is a community-developed list of software weakness types to better delineate and organize weaknesses around concepts frequently encountered in software development.

2 Findings Overview

2.1 Project info and Contract address

Project Name: CoboSafe

Project URL: https://app.safe.global/share/safe-app?appUrl=https%3A%2F%2Fsafeapp.cobo.com%2F&chain=matic

Audit Time: 2022/12.12 - 2022/12.16

Language: solidity

Contract Name	Smart Contract Address in Ethereum
CoboSubSafeFactory.sol	https://etherscan.io/address/0x4fdff384f51bd5e128e53b09effed79a39fb654e
CoboSafeFactory.sol	https://etherscan.io/address/0xf007134763bf791697a190fba76e7df41f934123
CoboSubSafe.sol	https://etherscan.io/address/0xA6D137901061AdaC94624181d75D56013d85148C
CoboSafeModule.sol	https://etherscan.io/address/0x368C1C8A30c6A26B0dA0Cd515F1D1F861Eb47383
Contract Name	**Smart Contract Address in Polygon**
CoboSubSafeFactory.sol	https://polygonscan.com/address/0x8d765371024ae481bfb850f53265dae8398945b1
CoboSafeFactory.sol	https://polygonscan.com/address/0x6dB0BB8BE0bd510120B46DFbA731FFDC7Bea3ab0
CoboSubSafe.sol	https://polygonscan.com/address/0x8281ad858d96765efb8dcba2e71ce928cd82cbe2
CoboSafeModule.sol	https://polygonscan.com/address/0x09ca9a159845d40ab2a5cf42079a3672c5e33035

Contract Name	**Smart Contract Address in Avax**
CoboSubSafeFactory.sol	https://snowtrace.io/address/0x09ca9a159845d40ab2a5cf42079a3672c5e33035
CoboSafeFactory.sol	https://snowtrace.io/address/0x8d765371024ae481bfb850f53265dae8398945b1
CoboSubSafe.sol	https://snowtrace.io/address/0x91b1d462f03752aac492e98673fa95fef0e21a51
CoboSafeModule.sol	https://snowtrace.io/address/0x8281ad858d96765efb8dcba2e71ce928cd82cbe2
Contract Name	**Smart Contract Address in BSC**
CoboSubSafeFactory.sol	https://bscscan.com/address/0x0c331b57af29a0196a30ec53c0b46db81e080c4a
CoboSafeFactory.sol	https://bscscan.com/address/0x91b1d462f03752aac492e98673fa95fef0e21a51
CoboSubSafe.sol	https://bscscan.com/address/0x3856b384e6066f4269abe4901087201cca1f0985
CoboSafeModule.sol	https://bscscan.com/address/0x9d32d826e5ef81bf3d5054b1035afad8570d69cd
Contract Name	**Smart Contract Address in Optimisim**
CoboSubSafeFactory.sol	https://optimistic.etherscan.io/address/0x9d32d826e5ef81bf3d5054b1035afad8570d69cd
CoboSafeFactory.sol	https://optimistic.etherscan.io/address/0x0c331b57af29a0196a30ec53c0b46db81e080c4a
CoboSubSafe.sol	https://optimistic.etherscan.io/address/0xff143c7abc18341f80746e67aa18417889c7531b

CoboSafeModule.sol	https://optimistic.etherscan.io/address/0x3856b384e6066f4269abe4901087201cca1f0985
Contract Name	**Smart Contract Address in Arbitrum**
CoboSubSafeFactory.sol	https://arbiscan.io/address/0x9d32d826e5ef81bf3d5054b1035afad8570d69cd
CoboSafeFactory.sol	https://arbiscan.io/address/0x0c331b57af29a0196a30ec53c0b46db81e080c4a
CoboSubSafe.sol	https://arbiscan.io/address/0xff143c7abc18341f80746e67aa18417889c7531b
CoboSafeModule.sol	https://arbiscan.io/address/0x3856b384e6066f4269abe4901087201cca1f0985
Contract Name	**Smart Contract Address in Gnosis chain(Xdai)**
CoboSubSafeFactory.sol	https://gnosisscan.io/address/0x9d32d826e5ef81bf3d5054b1035afad8570d69cd
CoboSafeFactory.sol	https://gnosisscan.io/address/0x0c331b57af29a0196a30ec53c0b46db81e080c4a
CoboSubSafe.sol	https://gnosisscan.io/address/0xff143c7abc18341f80746e67aa18417889c7531b
CoboSafeModule.sol	https://gnosisscan.io/address/0x3856b384e6066f4269abe4901087201cca1f0985

2.2 Summary

Severity	Found	
Critical	0	

High	0	
Medium	2	🟧 🟧
Low	0	
Informational	0	

2.3 Key Findings

Two Medium severities findings are related to owner authority, centralized risk.

ID	Severity	Findings Title	Status	Confirm
NVE-001	Medium	Owner has higher authority	Ignore	Confirmed
NVE-002	Medium	Owner has higher authority	Ignore	Confirmed

Table 2.1: Key Audit Findings

3 Detailed Description of Findings

3.1 Owner has higher authority

ID: NVE-001

Location: CoboSubSafeFactory.sol

Severity: Medium

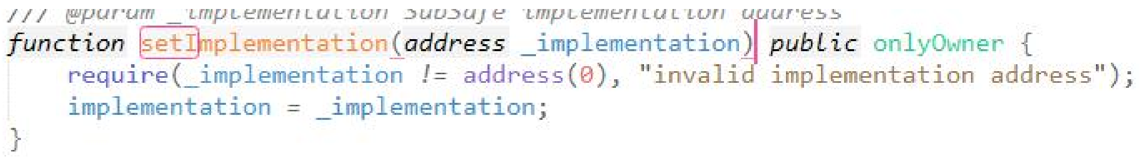
Category: Authority Issues

Likelihood: Medium

Impact: Medium

Description:

The CoboSubSafeFactory contract is a factory contract. Create a SubSafe contract by calling createSubSafe, and create it with the implementation template. However, the owner of the CoboSubSafeFactory contract can change the template for creating SubSafe, and find out that the owner is an eoa address through the data on the chain. If the owner changes the implementation of SubSafe, it will create code that does not match the description or add a backdoor function. The specific code segment is shown in the Figure 1.

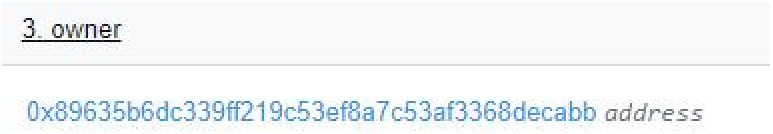


```
/// @param _implementation SubSafe implementation address
function setImplementation(address _implementation) public onlyOwner {
    require(_implementation != address(0), "invalid implementation address");
    implementation = _implementation;
}
```

Figure 1 function setImplementation

Eoa address:

https://polygonscan.com/address/0x89635b6dc339ff219c53ef8a7c53af3368decabb



3. owner

0x89635b6dc339ff219c53ef8a7c53af3368decabb *address*

Figure 2 Owner of CoboSubSafeFactory

Recommendations:

Numen Cyber Lab recommends proper management of private keys or use Gnosis multisig.

Result: Pass

Fix Result:

Ignore (After communicating with the project party, it will be changed to Gnosis multi-signature in the future.)

3.2 OWNER HAS HIGHER AUTHORITY

ID: NVE-002 Location:CoboSafeFactory.sol

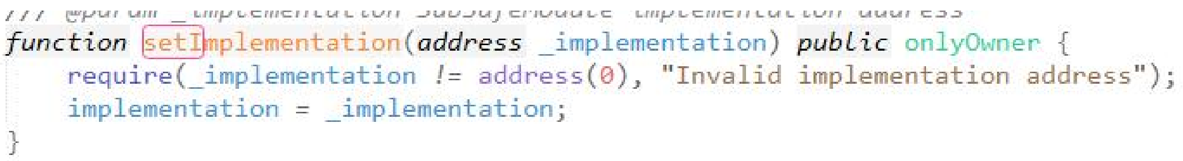
Severity: Medium Category: Authority Issues

Likelihood: Medium

Impact: Medium

Description:

The CoboSafeFactory contract creates a CoboSafeModule by calling createSafeModuleWithNonce. CoboSafeModule will create a CoboSafeModuleBase, which is the core part. CoboSafeModuleBase defines the granularity of multi-signature control to the function name and parameters, it sets different roles for permission control. Create CoboSafeModule by using implementation as a code template. The owner of the CoboSafeFactory contract can change the implementation of CoboSafeModule by calling setImplementaion, then the owner turns out to be an eoa.Considering if the owner changes the implementation of CoboSafeModule. This will create code that does not match the description or add a backdoor function. The specific code segment is shown in the Figure 3.

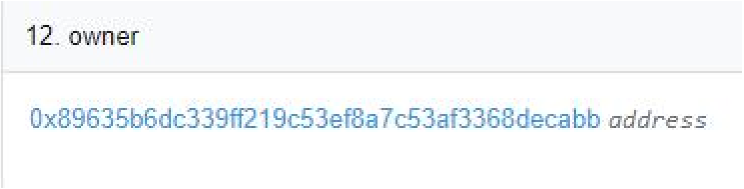


```
/// @param _implementation SubSafeModule implementation address
function setImplementation(address _implementation) public onlyOwner {
    require(_implementation != address(0), "Invalid implementation address");
    implementation = _implementation;
}
```

Figure 3 function setimplementation

Eoa address:

https://polygonscan.com/address/0x89635b6dc339ff219c53ef8a7c53af3368decabb



12. owner

0x89635b6dc339ff219c53ef8a7c53af3368decabb *address*

Figure 4 Owner of CoboSafeFactory

Recommendations:

Numen Cyber Lab recommends proper management of private keys or use Gnosis multisig.

Result: Pass

Fix Result:

Ignore(After communicating with the project party, it will be changed to Gnosis multi-signature in the future.)

4 Conclusion

In this audit, we thoroughly analyzed CoboSafe smart contract implementation. The problems found are described and explained in detail in Section 3. The problems found in the audit have been brought up to the project party, ignored issues are in line with the project design, and permissions are only used for the project to properly function. We therefore deem the audit result to be a **PASS.** To improve this report, we greatly appreciate any constructive feedbacks or suggestions, on our methodology, audit findings, or potential gaps in scope/coverage.

5 Appendix

5.1 Basic Coding Assessment

5.1.1 Apply Verification Control

- Description: The security of apply verification
- Result: Not found
- Severity: Critical

5.1.2 Authorization Access Control

- Description: Permission checks for external integral functions
- Result: Not found
- Severity: Critical

5.1.3 Forged Transfer Vulnerability

- Description: Assess whether there is a forged transfer notification vulnerability in the contract
- Result: Not found
- Severity: Critical

5.1.4 Transaction Rollback Attack

- Description: Assess whether there is transaction rollback attack vulnerability in the contract.
- Result: Not found
- Severity: Critical

5.1.5 Transaction Block Stuffing Attack

- Description: Assess whether there is transaction blocking attack vulnerability.
- Result: Not found
- Severity: Critical

5.1.6 soft fail Attack Assessment

- Description: Assess whether there is soft fail attack vulnerability.
- Result: Not found
- Severity: Critical

5.1.7 hard fail Attack Assessment

- Description: Examine for hard fail attack vulnerability
- Result: Not found
- Severity: Critical

5.1.8 Abnormal Memo Assessment

- Description: Assess whether there is abnormal memo vulnerability in the contract.
- Result: Not found
- Severity: Critical

5.1.9 Abnormal Resource Consumption

- Description: Examine whether abnormal resource consumption in contract processing.
- Result: Not found
- Severity: Critical

5.1.10 Random Number Security

- Description: Examine whether the code uses insecure random number.
- Result: Not found
- Severity: Critical

5.2 Advanced Code Scrutiny

5.2.1 Cryptography Security

- Description: Examine for weakness in cryptograph implementation.
- Results: Not Found
- Severity: High

5.2.2 Account Permission Control

- Description: Examine permission control issue in the contract
- Results: Not Found
- Severity: Medium

5.2.3 Malicious Code Behaviour

- Description: Examine whether sensitive behaviour present in the code
- Results: Not found
- Severity: Medium

5.2.4 Sensitive Information Disclosure

- Description: Examine whether sensitive information disclosure issue present in the code.
- Result: Not found
- Severity: Medium

5.2.5 System API

- Description: Examine whether system API application issue present in the code
- Results: Not found
- Severity: Low

6 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 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 without Numen's prior written consent.

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 Numen 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. Numen's position is that each company and individual are responsible for their own due diligence and continuous security. Numen'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.

REFERENCES

[1] MITRE. CWE- 191: Integer Underflow (Wrap or Wraparound). https://cwe.mitre.org/data/ definitions/191.html.

[2] MITRE. CWE- 197: Numeric Truncation Error. https://cwe.mitre.org/data/definitions/197. html.

[3] MITRE. CWE-400: Uncontrolled Resource Consumption. https://cwe.mitre.org/data/ definitions/400.html.

[4] MITRE. CWE-440: Expected Behavior Violation. https://cwe.mitre.org/data/definitions/440. html.

[5] MITRE. CWE-684: Protection Mechanism Failure. https://cwe.mitre.org/data/definitions/ 693.html.

[6] MITRE. CWE CATEGORY: 7PK - Security Features. https://cwe.mitre.org/data/definitions/ 254.html.

[7] MITRE. CWE CATEGORY: Behavioral Problems. https://cwe.mitre.org/data/definitions/438. html.

[8] MITRE. CWE CATEGORY: Numeric Errors. https://cwe.mitre.org/data/definitions/189.html.

[9] MITRE. CWE CATEGORY: Resource Management Errors. https://cwe.mitre.org/data/ definitions/399.html.

[10] OWASP. Risk Rating Methodology. https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology

NUMEN

Numen Cyber Technology Pte. Ltd.

11 North Buona Vista Drive, #04-09,
The Metropolis, Singapore 138589

Tel: 65-63555555

Fax: 65-63666666

Email: sales@numencyber.com

Web: https://numencyber.com