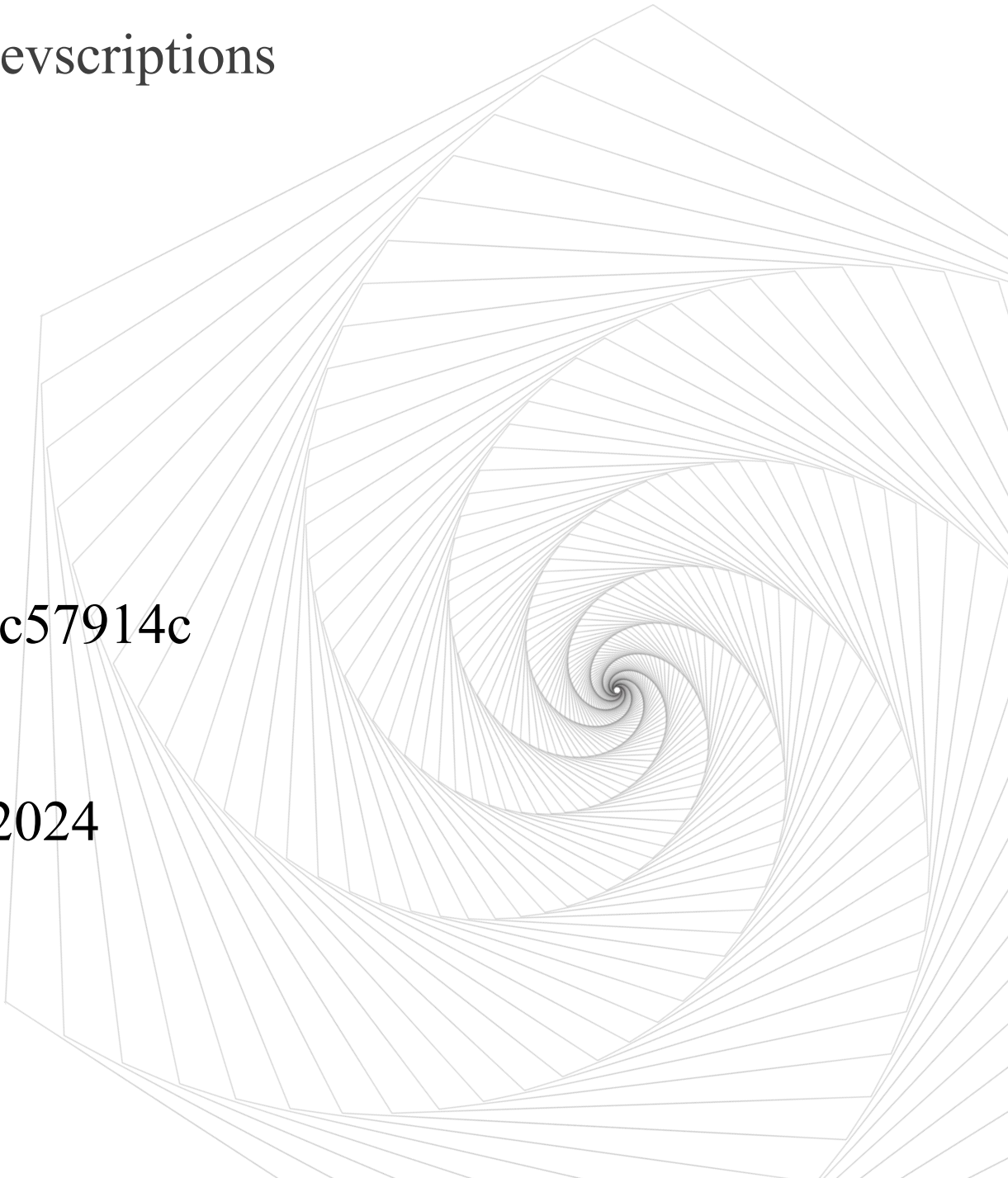
BINENET

Smart Contract Audit Report

BEVM Bevscriptions



V 1.0

SC.45ca7c57914c

Apr 3rd, 2024

Table Of Content

1 Report Overview

Binenet security team have audited the Bevscriptions, 0 risks was identified in Bevscriptions. users should pay attention to the following aspects when interacting with this project.

Contract Code	Function	Security Level	Status	Fix Result
BevscriptionsMarket.sol	batchMatchOrders	Remind	Audited	---
BevscriptionsMarket.sol	cancelOrders	Remind	Audited	---
BevscriptionsMarket.sol	refundOrders	Remind	Audited	---

***Risk Description: ---**

2 Asset Management Security Assessment

Asset Type	Function	Security Level
User Mortgage Token Assets	---	---
Users Mortgage Platform Currency Assets	---	---

***Description:** Check the management security of digital currency assets transferred by users in the contract business logic. Observe whether there are security risks that may cause the loss of customer funds, such as the digital currency assets transferred into the contract are incorrectly recorded or transferred out by mistake.

3 Audit Overview

3.1 Project Information

Bevscriptions is an inscription trading market on the BEVM.

This is a reference implementation of the Bevscriptions standard.

3.2 Audit Information

Project Name	Bevscriptions
Platform	BEVM
Audit Scope	BevscriptionsMarket.sol#aa8c6f9f8179891f1b28edc49d4fe715 SafeTransferLib.sol#efd93fb42678066a700f01fa86a209f8
Website	https://www.bevscriptions.com/

3.3 External Visibility Analysis

Function	Visibility	State Change	Modifier	Payable	Description
batchMatchOrders	public	true	nonReentrant	payable	---
cancelOrder	public	true	---	---	---
cancelOrders	public	true	---	---	---
disableAllFeatures	external	true	onlyOwner	---	---
disableFeature	public	true	onlyOwner	---	---
enableAllFeatur	external	true	onlyOwne	---	---

es			r		
enableFeature	public	true	onlyOwner	---	---
executeOrder	public	true	---	payable	---
refund	public	true	---	---	---
refundOrders	public	true	---	---	---
renounceOwnership	public	true	onlyOwner	---	---
setFeeAddress	external	true	onlyOwner	---	---
setFeeBps	external	true	onlyOwner	---	---
transferOwnership	public	true	onlyOwner	---	---
updateTrustedVerifier	external	true	onlyOwner	---	---

3.4 Audit Process

Audit time: 2024.4.3- 2024.4.3

Audit methods: Static Analysis, Dynamic Testing, Typical Case Testing and Manual Review.

Audit team: Binenet Security Team.

4 Security Finding Details

4.1 Redundant code

Severity Level : Remind

Lines : BevscriptionsMarket.sol # L94

Description: The feature determination of batchMatchOrders is redundant, and the internal function _executeOrder has determined the feature attribute. According to gas optimization principles, redundant code can be considered for removal.

```
ftrace | funcSig
93    function batchMatchOrders(BM20Order[] calldata orders⬆) public payable nonReentrant {
94        if (!featureIsEnabled['buy']) revert FeatureDisabled();
95
```

```
ftrace | funcSig
123   function _executeOrder(BM20Order calldata order⬆, uint256 userBalance⬆) internal {
124       if (!featureIsEnabled['buy']) revert FeatureDisabled();
```

Recommendations: Adjust according to actual situation.

Status : Audited.

Fix Result: Adjust according to actual situation.

4.2 Redundant code

Severity Level : Remind

Lines : BevscriptionsMarket.sol # L155

Description: The feature determination of cancelOrders is redundant, and the low-level function cancelOrder has determined the feature attribute. According to gas optimization principles, redundant code can be considered for removal.

```
ftrace | funcSig
154     function cancelOrders(BM20Order[] calldata orders⬆) public {
155         if (!featureIsEnabled['cancel']) revert FeatureDisabled();
156
157         for (uint8 i = 0; i < orders⬆.length; i++) {
158             BM20Order calldata order = orders⬆[i];
159
160             cancelOrder(order);
161         }
162     }
ftrace | funcSig
163     function cancelOrder(BM20Order calldata order⬆) public {
164         if (!featureIsEnabled['cancel']) revert FeatureDisabled();
165
```

Recommendations: Adjust according to actual situation.

Status : Audited.

Fix Result: Adjust according to actual situation.

4.3 Redundant code

Severity Level : Remind

Lines : BevscriptionsMarket.sol # L181

Description: The feature determination of refundOrders is redundant, and the low-level function refund has determined the feature attribute. According to gas optimization principles, redundant code can be considered for removal.

```
ftrace | funcSig
180     function refundOrders(BM20Order[] calldata orders⬆) public {
181         if (!featureIsEnabled['cancel']) revert FeatureDisabled();
182
183         for (uint8 i = 0; i < orders⬆.length; i++) {
184             BM20Order calldata order = orders⬆[i];
185
186             refund(order);
187         }
188     }
ftrace | funcSig
189     function refund(BM20Order calldata order⬆) public {
190         if (!featureIsEnabled['cancel']) revert FeatureDisabled();
191
```

Recommendations: Adjust according to actual situation.

Status : Audited.

Fix Result: Adjust according to actual situation.

5 Audit Categories

Categories	Subitems
Business Security	Transfer token function
	Mint token and burn token vulnerability
	Contract logic function
	Mining pool deposit and withdrawal function
	Reasonableness of agreement amendment
	Functional design
	Dos caused by time
	Insecure oracles and their design
	Deployer private key leak hazard
General Vulnerability	Compiler version security
	Redundant code
	Use of safemath library
	Not recommended encoding
	Use require/assert mistakely
	Fallback function safety
	tx.origin authentication
	Owner permission control
	Gas consumption detection
	Call injection attack
	Low-level function safety
	Additional token vulnerabilities
	Access control
	Numeric overflow detection
	Arithmetic precision error

	Misuse of random number detection
	Unsafe external call
	Variable override
	Uninitialized storage pointer
	Return value call validation
	Transaction order dependent detection
	Timestamp dependent attack
	Denial of service attack detection
	Fake recharge vulnerability detection
	Reentrancy Attack Detection
	Replay attack detection
	Reordering attack detection

6 Explanation Of Vulnerability Rating

Vulnerability Rating	Rating Description
High Risk Vulnerability	Vulnerabilities that can directly cause the loss of token contracts or user funds, such as: overflow、reentrancy 、false recharge， which can cause the value of tokens to be zeroed, or causing false exchanges to lose tokens, or causing losing ETH or tokens, etc; Vulnerabilities that can cause loss of ownership of token contracts, such as: access control flaws of key functions, call injection leading to access control bypass of key functions, etc; Vulnerabilities that can cause token contracts to fail to work properly, such as: denial of service vulnerabilities caused by sending ETH to malicious addresses, and denial of service vulnerabilities caused by gas exhaustion;
Medium Risk Vulnerability	High-risk vulnerabilities that require specific addresses to be triggered, such as overflow that can only be triggered by token contract owners; access control flaws of non-critical functions, logic design flaws that cannot cause direct financial losses, etc;
Low Risk Vulnerability	Vulnerabilities that are difficult to be triggered, vulnerabilities that cause limited harm after triggering, such as overflow vulnerabilities that require a large amount of ETH or tokens to be triggered, vulnerabilities that the attacker cannot directly profit after triggering overflow, and transaction sequence-dependent risks

	triggered by specifying high gas wait;

7 Statement

Binenet only issues this report based on the facts that have occurred or existed before the issue of this report, and assumes corresponding responsibilities for it. For the facts that occurred or existed after the issuance, we cannot judge the security status of the smart contract , and we will not be responsible for it.

This report does not include external contract calls , new types of attacks that may appear in the future, and contract upgrades or tampered codes (with the development of the project side, smart contracts may add new pools, new functional modules, new external contract calls, etc.), does not include front-end security and server security.

The documents and materials provided to us by the information provider as of the date of this report.

Binenet assumes that there is no missing, tampered, deleted or concealed information provided. If the information provided is missing, tampered, deleted, concealed or reflected inconsistent with the actual situation, Binenet shall not be liable for any losses and adverse effects resulting therefrom.

8 About Binenet

Founded in June 2021, Binenet is a dedicated and pure blockchain security company, focusing on accurate, efficient and intelligent blockchain threat detection and response. Committed to providing users with professional products and dedicated services in the field of blockchain security. Business functions cover penetration testing, code auditing, emergency response, on-chain data monitoring, AML anti-money laundering, etc., covering all aspects of blockchain ecosystem security.

Official Website
https://binenet.com

Telegram
https://t.me/binenetxyz

Twitter
https://twitter.com/binenetxyz

E-mail
team@binenet.com