BINENET

Smart Contract Audit Report

BNBChain

BiYing Project

V 2.0

SC.209812817909

Nov. 21st, 2025

Table Of Content

1 Report Overview

Binenet security audit has identified 0 significant security issues across the BiYing smart contracts, with 0 High-severity, 0 Medium-severity, and 0 Low-severity vulnerabilities.

Contract Code	Function	Security Level	Status	Fix Result
BY.sol:L107	---	Info	Audited	Known

***Risk Description:** No obvious security issues found based on business logic.

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

BIYing was built on the simple belief that finance should be open, transparent, and governed by code. It's not just a product; it's a declaration of financial freedom through blockchain technology.

BIYing is a fully decentralized on-chain wealth management platform that revolutionizes traditional finance through smart contract automation. The platform offers:

Fully Decentralized On-chain Wealth Management: All operations are transparent and executed automatically on the blockchain

Public Compound Interest: Returns are publicly disclosed for 1-day, 15-day, and 30-day staking periods

Transparency, Decentralization, Automated On-chain Execution: Ensuring fairness and security through immutable smart contracts

This is a reference implementation of the BiYing standard. Built on Solidity with fee management, staking rewards, node dividends, and multi-level referral incentives.

3.2 Audit Information

Project Name	BiYing
Platform	BNBChain
Audit Scope	BY Token: https://bscscan.com/address/0xd4713664B4997299bB41273432a77FbB44eED6DC BYB Token: https://bscscan.com/address/0xd1c5840E565f7350A893cd036367f639CB66B75f Referral:

	https://bscscan.com/address/0x80472Ca15f2B6a3Ba29E18952e180D859C20e611 Staking: https://bscscan.com/address/0xB7dab22F4Bc9Bc6eC24cefeFd846c00d401676D1 SharePool: https://bscscan.com/address/0x35a9dA35B90A3e532d13Be7e6936088a79724B65
Website	https://biying.fun/

3.3 External Visibility Analysis

Function	Visibility	State Change	Modifier	Payable	Description
BY Token					
setSwapAtAmount	public	true	onlyOwner	---	---
setMarketingAddress	external	true	onlyOwner	---	---
setProfitAddress	external	true	onlyOwner	---	---
setStakingAddress	external	true	onlyOwner	---	---
setSharePoolAddress	external	true	onlyOwner	---	---
recycle	external	true	---	---	---
Staking					
start	external	true	onlyOwner	---	---
stop	external	true	onlyOwner	---	---
setByAddress	external	true	onlyOwner	---	---
initNodes	public	true	onlyOwner	---	---
setMaxLevels	external	true	onlyOwner	---	---
setSharePoolAddress	external	true	onlyOwner	---	---

setReferralAddress	external	true	onlyOwner	---	---
setTeamVirtuallyInvestValue	external	true	onlyOwner	---	---
setMarketingAddress	external	true	onlyOwner	---	---
stakeWithInviter	external	true	onlyEOA	---	---
stake	public	true	---	---	---
unstake	external	true	onlyEOA	---	---
sync	external	true	---	---	---
emergencyWithdrawLAF	external	true	onlyOwner	---	---
Referral					---
setStakingContract	external	true	onlyOwner	---	---
transferOwnership	external	true	onlyOwner	---	---
bindReferral	external	true	onlyStakingContract	---	---
SharePool					
transferOwnership	public	true	onlyOwner	---	---
setByContract	external	true	onlyOwner	---	---
setStakingContract	external	true	onlyOwner	---	---
addPreacherCount	external	true	onlyStakingContract	---	---
subPreacherCount	external	true	onlyStakingContract	---	---
harvest	external	true	onlyStakingContract	---	---
harvest	external	true	---	---	---
initNodes	public	true	onlyOwner	---	---
removeNode	public	true	onlyOwner	---	---
updatePool	external	true	onlyByContract	---	---

3.4 Audit Process

Audit time: November 9, 2025 - November 21, 2025

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

Audit team: Binenet Security Team.

4 Audit Categories

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

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

5 Security Finding Details

5.1 Price Manipulation Vulnerability in BY._transfer

Severity Level : Info

Lines : BY.sol:L107

Description:

In BY.sol at line 107, the _transfer function checks if getPrice() >= 10 ether before allowing buy transactions. The getPrice() function (line 70-73) directly reads from Uniswap pair reserves without any protection mechanisms: price = (reserveUSDT * 1e18) / reserveBY. This creates a critical vulnerability:

1) Flash Loan Attack: An attacker can use a flash loan to temporarily manipulate the Uniswap pair reserves by making a large trade, causing getPrice() to return a manipulated value above 10 ether, then execute a buy transaction at the manipulated price, and repay the flash loan.

2) Price Oracle Manipulation: Since the price is read directly from the DEX pair without TWAP (Time-Weighted Average Price) or other protection mechanisms, a single large trade can significantly alter the price calculation.

3) Bypass Protection: Attackers can bypass the minimum price protection to buy tokens at artificially low prices.

4) Front-running: Malicious actors can front-run legitimate transactions by manipulating the price just before the transaction executes. The vulnerability is particularly dangerous because it affects the core buy mechanism and can lead to significant financial losses.

```
ftrace | funcSig
70    function getPrice() public view returns (uint256 price) {
71        (uint112 reserveUSDT, uint112 reserveBY) = getReserves();
72        price = (uint256(reserveUSDT) * 1e18) / uint256(reserveBY);
73    }
74
```

Recommendations:

1. Implement TWAP (Time-Weighted Average Price) oracle instead of using spot price
2. Use multiple price sources and take the median/average.
3. Ref: https://docs.uniswap.org/contracts/v2/concepts/core-concepts/oracles

Status *: Audited.*

Fix Result: ***Known.***

5.2 Compound Interest Calculation Mechanism

Severity Level **:** Pass

Lines **:** Staking#L445-L459, Staking#L23-L27

Description**:** The compound interest calculation uses PRB Math library's UD60x18 fixed-point arithmetic for precise decimal calculations. The *caclItem()* function calculates rewards based on staking period and type-specific rates (0.3% daily for 1-day, 0.6% daily for 15-day, 1.2% daily for 30-day staking). The calculation formula uses *stakingAmount * rate^stake_period* with a maximum period cap of 30 days. The implementation ensures safe multiplication and exponentiation operations through the PRB Math library, preventing overflow and precision loss.

Recommendations: ---

Status : *Audited.*

Fix Result: ---

5.3 Referral Reward Distribution System

Severity Level **:** Pass

Lines **:** Staking#L512-L527, Staking#L625-L637, Staking#L639-L725

Description**:** The referral system implements a two-tier reward mechanism: direct referral rewards (5% of profit) and team differential rewards (up to 20% based on KPI levels). The *directReferralReward()* function verifies the referrer is a "Preacher" (staked

≥100 BY tokens) before distribution. The *teamReward()* function implements a differential reward system across 30 referral levels with KPI-based tiers (S1: 4%, S2: 8%, S3: 12%, S4: 16%, S5: 20%). Reward calculations use safe division operations, and unclaimed rewards are automatically transferred to the marketing address to prevent fund loss.

Recommendations: ---

Status : *Audited.*

Fix Result: ---

5.4 Node Activation and SharePool Distribution

Severity Level : Pass

Lines: SharePool#L74-L96, SharePool#L98-L127, SharePool#L137-L150, SharePool#L165-L167

Description: The node activation mechanism requires nodes to hold 500 BY tokens and have at least 10 direct "Preachers" (users with ≥100 BY staked). The *enableNode()* and *disableNode()* functions properly manage node state transitions with reward debt tracking. The *updatePool()* function distributes rewards proportionally to all active nodes using the *shareUnitAcc* accumulator pattern, ensuring fair distribution. The *_harvest()* function calculates pending rewards using *shareUnitAcc* - *rewardDebt* pattern, preventing double-spending and ensuring accurate reward accounting.

Recommendations: ---

Status : *Audited.*

Fix Result: ---

5.5 Automatic Token Burn Mechanism

Severity Level : Pass

Lines : BY#L94-L102, Staking#L254-L266, Staking#L497

Description: The token burn mechanism automatically burns 1% of tokens on each

buy and sell transaction until 1,000,000 tokens are burned. The *_transfer*() function in BY contract checks the total burned amount against the target (1,000,000 tokens) before executing burns. During staking redemption, the *buyAndBurn*() function swaps USDT for BY tokens and sends them to the dead address (0xdead). The burn mechanism uses safe subtraction operations and prevents burns beyond the target amount, creating deflationary pressure on the token supply.

Recommendations: ---

Status : *Audited.*

Fix Result: ---

5.6 Price Protection and Fee Distribution

Severity Level : Pass

Lines : BY#L106-L109, BY#L80-202, BY#L214-242

Description: The contract implements price protection by preventing buys when the price falls below 10 USDT per BY token, protecting investors from significant price drops. The *_transfer()* function handles fee distribution for buy transactions (1% burn, 1% LP, 1% node share, 2% dev fee) and sell transactions (1% burn, 1% node share, 3% marketing fee). The *swapTokenForFund()* function accumulates fees and swaps them when reaching the threshold (*swapAtAmount*), ensuring efficient gas usage. Fee distribution uses safe arithmetic operations and proper access control through modifiers, preventing unauthorized fee manipulation.

Recommendations: ---

Status : *Audited.*

Fix Result: ---

5.7 Contract Access Control and Owner Management

Severity Level : Pass

Lines :

BY#L338-360, Staking#L128-L162, SharePool#L54-L161, Referral#L29-L51, ExcludedFromFeeList#L16-L26

***Description*:** All contracts implement proper access control mechanisms through the onlyOwner modifier to protect critical administrative functions. The BY contract provides owner-only functions for managing fee parameters and contract addresses: setSwapAtAmount() (BY#L338-340) allows adjusting the fee accumulation threshold for gas optimization, while address management functions setMarketingAddress(), setProfitAddress(), setStakingAddress(), and setSharePoolAddress() (BY#L342-360) automatically add new addresses to the fee exclusion list. The Staking contract implements owner controls for system configuration including start() and stop() functions to enable or disable staking operations, setMaxLevels() for referral level adjustments, and address management functions for BY, SharePool, Referral, and Marketing contracts (Staking#L128-L162). The SharePool contract allows owner to initialize and remove nodes, and configure contract addresses (SharePool#L54-L161). The Referral contract provides owner functions for setting the staking contract address and transferring ownership (Referral#L29-L51). The ExcludedFromFeeList contract enables owner to manage fee exclusion lists for multiple accounts (ExcludedFromFeeList#L16-L26). All owner functions are protected by the onlyOwner modifier inherited from the Owned contract (solmate library), ensuring that only the contract owner can modify these critical parameters. The ownership transfer mechanism allows for secure ownership transitions when needed. This design provides necessary operational flexibility while maintaining security through proper access control, preventing unauthorized modifications to critical contract parameters.

Recommendations: ---

Status : *Audited.*

Fix Result: ---

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