SOURCEHAT

888Bits

Smart Contract Audit Report

EXECUTIVE SUMMARY

This report presents the outcomes of our collaborative engagement with the 888Bits team, focusing on the comprehensive evaluation of the 888Bits and S8BFeeWallet contracts.

Our team conducted an initial security assessment from **June 19th** to **June 21st, 2024**.

888Bits is developing a new ERC-20 token that functions as an automatic liquidity-providing protocol to the S8B-USDC Pair. Additionally, the team can set up vesting schedules that lock tokens in the contract, allowing users to claim them gradually over a vesting period.

AUDIT SCOPE

Name	Source Code	Visualized
S8B	6eab59a	Inheritance Chart. Function Graph.

Name	Address/Source Code	Visualized
S8B	6eab59a	Inheritance Chart. Function Graph.

AUDIT FINDINGS

No findings were identified, though some centralized aspects are present.

SYSTEM OVERVIEW

DEPLOYMENT AND TOKENOMICS

The total supply of the token is set to ~888.89 million $S8B [888,888,888]. No mint functions are accessible beyond deployment. Any user can burn their own tokens to reduce the total supply at any time. A new S8B-USDC Pair and S8BFeeWallet contract are created

upon deployment. The deployer can specify a list of addresses that are automatically added to the contract's whitelist.

LAUNCH

The owner can execute the contract's launch process which will:

- Set the whitelist end time to 3 minutes in the future.
- Whitelist every address that currently owns a ProofPass NFT.
- Enable trading and record the timestamp in which trading is enabled.

TRANSFERS

Trading must be enabled before transfers can take place on the platform. Before the whitelist end time has been reached, both the sender and the recipient must have been added to the contract's whitelist in order for a transfer to successfully occur. Blacklisted accounts are prohibited from participating in transfers.

The contract enforces a maximum sell amount when the restrict-whales functionality is enabled by the owner, which imposes a limit to the number of tokens that can be sold via an Automated Market Maker Pair address in a single transaction.

The contract enforces a maximum wallet amount when the restrict-whales functionality is enabled by the owner, which prevents a transfer from occurring if the recipient's token balance exceeds the limit number of tokens after the transfer occurs.

There is a Liquidity fee, Rewards fee, Team fee, Dev fee, and Proof fee on all buys and sells via an approved Pair address where neither the

sender nor the recipient is excluded from fees. A separate fee structure can be set by the team to apply different fee percentages depending on whether the transaction is a buy or sell.

If at least 1 day but less than 31 days has passed since launch, the Proof fee will be automatically updated from 2% to 1% on both buys and sells and the Dev fee will remain zero. After 31 days have passed, the Proof fee will be permanently removed and the Dev fee will be updated from zero to 1% on both buys and sells.

The tokens collected through fees are stored in the contract address. The tokens are swapped for USDC for the purpose of funding Uniswap liquidity and designated addresses when the following conditions are met:

- The automatic liquidity add functionality is enabled by the team.
- The threshold number of tokens in the contract address (determined by the owner) has been reached.
- The contract is not currently performing an automatic liquidity add.
- The caller is not initiating a buy transaction via an approved Pair address.

Liquidity adds are automatically performed by selling the tokens collected as fees, pairing the received USDC with the token, and adding it as liquidity to the USDC pair. The LP tokens received through this process are sent to the 0x..dead address.

The tokens collected through the Rewards Fee and Team fee are swapped for USDC and sent to the team's Rewards wallet and Team wallet respectively. The tokens collected through the Proof fee are

swapped for USDC and evenly split between the Proof Revenue address and Proof wallet.

The contract complies with the ERC-20 standard.

TOKEN VESTING

The owner can create a new lock by specifying a list of beneficiary addresses, start time, duration, and token amount. The total number of tokens is transferred from the owner to the contract and the lock details are recorded in the platform.

Any user can specify a beneficiary address to release vested tokens for once the lock start time associated with any of their created locks has passed. If the full lock duration has passed, the user will receive 100% of the tokens due to them. Otherwise, the released amount is proportional to the elapsed time since the lock start time. Vested tokens are transferred from the contract to the beneficiary, and lock details are updated or removed if fully released.

OWNERSHIP CONTROLS

The owner can update the Liquidity fee, Rewards fee, Team fee for both the buy and sell fee structures at any time. The total fee percentages combined (including the current Proof fee and Dev fee) cannot exceed 12% on buys and 17% on sells. If at least 1 day but less than 31 days has passed since launch, the Proof fee will be automatically updated from 2% to 1% on both buys and sells and the Dev fee will remain zero. After 31 days have passed, the Proof fee will be permanently removed and the Dev fee will be updated from zero to 1% on both buys and sells. The owner can include or exclude accounts from fees at any time.

The owner can add or remove any address from the contract's blacklist at any time. The owner can pause or unpause trading at any time. The owner can initiate a transfer to multiple users in a single transaction by specifying a list of addresses and corresponding token amounts.

The owner can specify a list of addresses to add to the contract's whitelist at any time. The owner can set the maximum sell amount and maximum wallet amount to any values at any time. The owner can include or exclude accounts from the maximum sell and maximum wallet restrictions at any time. The owner can enable or disable the maximum sell and maximum wallet restrictions at any time.

The owner can enable/disable automatic liquidity adds at any time. The owner can update the threshold number of tokens needed to trigger an automatic liquidity add to any value at any time.

The owner can withdraw any tokens not collected through fees from the contract at any time. The owner can add or remove any address as an Automated Market Maker Pair at any time. The owner can update the Fee wallet address referenced in the contract at any time. The owner can set the team's Rewards wallet and Team wallet to any addresses at any time. The owner can transfer ownership to any address at any time.

VULNERABILITY ANALYSIS

Vulnerability Category	Notes	Result
Arbitrary Jump/Storage Write	N/A	PASS
Centralization of Control	◦ The owner can add any address to the contract's blacklist at any time. ◦ The owner can pause trading at any time.	WARNING
Compiler Issues	N/A	PASS
Delegate Call to Untrusted Contract	N/A	PASS
Dependence on Predictable Variables	N/A	PASS
Ether/Token Theft	N/A	PASS
Flash Loans	N/A	PASS

Vulnerability Category	Notes	Result
Front Running	N/A	PASS
Improper Events	N/A	PASS
Improper Authorization Scheme	N/A	PASS
Integer Over/Underflow	N/A	PASS
Logical Issues	N/A	PASS
Oracle Issues	N/A	PASS
Outdated Compiler Version	N/A	PASS
Race Conditions	N/A	PASS
Reentrancy	N/A	PASS
Signature Issues	N/A	PASS
Sybil Attack	N/A	PASS

Vulnerability Category	Notes	Result
Unbounded Loops	N/A	PASS
Unused Code	N/A	PASS
Overall Contract Safety		PASS

ABOUT SOURCEHAT

SourceHat has quickly grown to have one of the most experienced and well-equipped smart contract auditing teams in the industry. Our team has conducted 1800+ solidity smart contract audits covering all major project types and protocols, securing a total of over $50 billion U.S. dollars in on-chain value!
Our firm is well-reputed in the community and is trusted as a top smart contract auditing company for the review of solidity code, no matter how complex. Our team of experienced solidity smart contract auditors performs audits for tokens, NFTs, crowdsales, marketplaces, gambling games, financial protocols, and more!

Contact us today to get a free quote for a smart contract audit of your project!

WHAT IS A SOURCEHAT AUDIT?

Typically, a smart contract audit is a comprehensive review process designed to discover logical errors, security vulnerabilities, and optimization opportunities within code. A *SourceHat Audit* takes this a step further by verifying economic logic to ensure the stability of smart contracts and highlighting privileged functionality to create a report that is easy to understand for developers and community members alike.

HOW DO I INTERPRET THE FINDINGS?

Each of our Findings will be labeled with a Severity level. We always recommend the team resolve High, Medium, and Low severity findings prior to deploying the code to the mainnet. Here is a breakdown on what each Severity level means for the project:

- **High** severity indicates that the issue puts a large number of users' funds at risk and has a high probability of exploitation, or the smart contract contains serious logical issues which can prevent the code from operating as intended.
- **Medium** severity issues are those which place at least some users' funds at risk and has a medium to high probability of exploitation.
- **Low** severity issues have a relatively minor risk association; these issues have a low probability of occurring or may have a minimal impact.
- **Informational** issues pose no immediate risk, but inform the project team of opportunities for gas optimizations and following smart contract security best practices.

GO HOME

SOURCEHAT

888BitsStaking

Smart Contract Audit Report

EXECUTIVE SUMMARY

This report presents the outcomes of our collaborative engagement with the 888Bits team, focusing on the comprehensive evaluation of the 888BitsStaking contract. We previously reviewed the project team's S8B token here.

Our team conducted an initial security assessment from **June 12th** to **June 14th, 2024**.

888Bits Staking is a new contract which allows users to stake S8B in exchange for various rewards.

AUDIT SCOPE

Name	Source Code	Visualized
888BitsStaking	Code provided by project team	Inheritance Chart. Function Graph.

Name/Source Code	Visualized
888BitsStaking Code provided by project team	Inheritance Chart. Function Graph.

AUDIT FINDINGS

No findings were identified, though some centralized aspects are present.

SYSTEM OVERVIEW

STAKING & REWARD ACCRUAL

The staking mechanism offers users three options for stake durations: 7 days, 14 days, or 30 days. Each duration corresponds to distinct reward accrual settings, influencing how users earn rewards. Specifically, a 7-day stake earns S8B rewards solely for the duration of the stake, while a

14-day stake continues to accrue S8B rewards post the initial 14 days. Additionally, a 30-day stake not only continuously earns S8B rewards but also qualifies for USDC dividend rewards.

STAKE MANAGEMENT

Each user stake is treated independently, ensuring that creating a new stake does not impact existing stakes' rewards or unlock time. Users have the flexibility to unstake at any time after a stake's duration has passed. Any of the Stake's pending S8B rewards are automatically harvested upon unstaking, but pending S8B or dividend rewards can be claimed manually at any time. When manually claiming rewards, users can choose to compound their rewards, which adds them back into their existing Stake. A user's cumulative S8B rewards are all harvested at once; USDC rewards can be either cumulatively harvested or harvested for individual Stakes. 7 Day Stakes are not compounded if their end date has passed. When USDC rewards are compounded, they are first swapped for S8B tokens before being added back into the Stake. The user must provide a mininmum accepted S8B to receive from the swap to protect against front running.

REWARD ACCRUAL AND DISTRIBUTION

Rewards are accrued on a daily basis, with reward pool updates automatically triggered by any stake, unstake, reward injection, or claim. Pool updates can also be manually triggered by any address at any time. Rewards for multiple days are processed at once if more than one day has passed since the last update. Any address can supply S8B or USDC rewards for distribution, with S8B rewards offering the additional feature of being distributed evenly across a specified

number of days in the future. The project team cannot supply USDC rewards while there are no active 30-day stakers.

VULNERABILITY ANALYSIS

Vulnerability Category	Notes	Result
Arbitrary Jump/Storage Write	N/A	PASS
Centralization of Control	N/A	PASS
Compiler Issues	N/A	PASS
Delegate Call to Untrusted Contract	N/A	PASS
Dependence on Predictable Variables	N/A	PASS
Ether/Token Theft	N/A	PASS

Vulnerability Category	Notes	Result
Flash Loans	N/A	PASS
Front Running	N/A	PASS
Improper Events	N/A	PASS
Improper Authorization Scheme	N/A	PASS
Integer Over/Underflow	N/A	PASS
Logical Issues	N/A	PASS
Oracle Issues	N/A	PASS
Outdated Compiler Version	N/A	PASS
Race Conditions	N/A	PASS
Reentrancy	N/A	PASS
Signature Issues	N/A	PASS

Vulnerability Category	Notes	Result
Sybil Attack	N/A	PASS
Unbounded Loops	N/A	PASS
Unused Code	N/A	PASS
Overall Contract Safety		PASS

ABOUT SOURCEHAT

SourceHat has quickly grown to have one of the most experienced and well-equipped smart contract auditing teams in the industry. Our team has conducted 1800+ solidity smart contract audits covering all major project types and protocols, securing a total of over $50 billion U.S. dollars in on-chain value!
Our firm is well-reputed in the community and is trusted as a top smart contract auditing company for the review of solidity code, no matter how complex. Our team of experienced solidity smart contract auditors performs audits for tokens, NFTs, crowdsales, marketplaces, gambling games, financial protocols, and more!

Contact us today to get a free quote for a smart contract audit of your project!

WHAT IS A SOURCEHAT AUDIT?

Typically, a smart contract audit is a comprehensive review process designed to discover logical errors, security vulnerabilities, and optimization opportunities within code. A *SourceHat Audit* takes this a step further by verifying economic logic to ensure the

stability of smart contracts and highlighting privileged functionality to create a report that is easy to understand for developers and community members alike.

HOW DO I INTERPRET THE FINDINGS?

Each of our Findings will be labeled with a Severity level. We always recommend the team resolve High, Medium, and Low severity findings prior to deploying the code to the mainnet. Here is a breakdown on what each Severity level means for the project:

- **High** severity indicates that the issue puts a large number of users' funds at risk and has a high probability of exploitation, or the smart contract contains serious logical issues which can prevent the code from operating as intended.
- **Medium** severity issues are those which place at least some users' funds at risk and has a medium to high probability of exploitation.
- **Low** severity issues have a relatively minor risk association; these issues have a low probability of occurring or may have a minimal impact.
- **Informational** issues pose no immediate risk, but inform the project team of opportunities for gas optimizations and following smart contract security best practices.

GO HOME