SPYWOLF

Security Audit Report

Audit prepared for

SCORCH

Completed on

June 20, 2025

OVERVIEW

This goal of this report is to review the main aspects of the project to help investors make an informative decision during their research process.

You will find a a summarized review of the following key points:

- ✔ Contract's source code
- ✔ Owners' wallets
- ✔ Tokenomics
- ✔ Team transparency and goals
- ✔ Website's age, code, security and UX
- ✔ Whitepaper and roadmap
- ✔ Social media & online presence

The results of this audit are purely based on the team’s evaluation and does not guarantee nor reflect the projects outcome and goal

- SPYWOLF Team -

TABLE OF CONTENTS

SCORCH

PROJECT DESCRIPTION

SCORCH is built on Shibarium Layer-2. Its main launch feature is the conversion (or "burning") of SHIB in exchange for SCORCH tokens. This mechanism permanently removes SHIB from circulation.

As the project expands, it becomes a "burn platform" that any project owner or token holder can use to facilitate community-driven token burns.

Release Date: TBD

Category: Token

CONTRACT INFO

Token Name
SCORCH

Symbol
SCORCH

Contract Address
Testnet

Network
Shibarium

Language
Solidity

Deployment Date
Not Deployed Yet

Contract Type
Staking

Total Supply
15,000,000,000

Status
Not Deployed Yet

TAXES

Buy Tax
1%

Sell Tax
1%

Our Contract Review Process

The contract review process pays special attention to the following:

- Testing the smart contracts against both common and uncommon vulnerabilities
- 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.

Blockchain security tools used:

- OpenZeppelin
- Mythril
- Solidity Compiler
- Hardhat

TOKEN TRANSFERS STATS

Transfer Count	Not Deployed Yet
Uniq Senders	Not Deployed Yet
Uniq Receivers	Not Deployed Yet
Total Amount	Not Deployed Yet
Median Transfer Amount	Not Deployed Yet
Average Transfer Amount	Not Deployed Yet
First transfer date	Not Deployed Yet
Last transfer date	Not Deployed Yet
Days token transferred	Not Deployed Yet

SMART CONTRACT STATS

Calls Count	Not Deployed Yet
External calls	Not Deployed Yet
Internal calls	Not Deployed Yet
Transactions count	Not Deployed Yet
Uniq Callers	Not Deployed Yet
Days contract called	Not Deployed Yet
Last transaction time	Not Deployed Yet
Created	Not Deployed Yet
Create TX	Not Deployed Yet
Creator	Not Deployed Yet

VULNERABILITY ANALYSIS

ID	Title	
SWC-100	Function Default Visibility	**Passed**
SWC-101	Integer Overflow and Underflow	**Passed**
SWC-102	Outdated Compiler Version	**Passed**
SWC-103	Floating Pragma	**Passed**
SWC-104	Unchecked Call Return Value	**Passed**
SWC-105	Unprotected Ether Withdrawal	**Passed**
SWC-106	Unprotected SELFDESTRUCT Instruction	**Passed**
SWC-107	Reentrancy	**Passed**
SWC-108	State Variable Default Visibility	**Passed**
SWC-109	Uninitialized Storage Pointer	**Passed**
SWC-110	Assert Violation	**Passed**
SWC-111	Use of Deprecated Solidity Functions	**Passed**
SWC-112	Delegatecall to Untrusted Callee	**Passed**
SWC-113	DoS with Failed Call	**Passed**
SWC-114	Transaction Order Dependence	**Passed**
SWC-115	Authorization through tx.origin	**Passed**
SWC-116	Block values as a proxy for time	**Passed**
SWC-117	Signature Malleability	**Passed**
SWC-118	Incorrect Constructor Name	**Passed**

VULNERABILITY ANALYSIS

ID	Title	
SWC-119	Shadowing State Variables	Passed
SWC-120	Weak Sources of Randomness from Chain Attributes	Passed
SWC-121	Missing Protection against Signature Replay Attacks	Passed
SWC-122	Lack of Proper Signature Verification	Passed
SWC-123	Requirement Violation	Passed
SWC-124	Write to Arbitrary Storage Location	Passed
SWC-125	Incorrect Inheritance Order	Passed
SWC-126	Insufficient Gas Griefing	Passed
SWC-127	Arbitrary Jump with Function Type Variable	Passed
SWC-128	DoS With Block Gas Limit	Passed
SWC-129	Typographical Error	Passed
SWC-130	Right-To-Left-Override control character (U+202E)	Passed
SWC-131	Presence of unused variables	Passed
SWC-132	Unexpected Ether balance	Passed
SWC-133	Hash Collisions With Multiple Variable Length Arguments	Passed
SWC-134	Message call with hardcoded gas amount	Passed
SWC-135	Code With No Effects	Passed
SWC-136	Unencrypted Private Data On-Chain	Passed

VULNERABILITY ANALYSIS

NO ERRORS FOUND

MANUAL
CODE REVIEW

When performing smart contract audits, our specialists look for known vulnerabilities as well as logical and access control issues within the code. The exploitation of these issues by malicious actors may cause serious financial damage to projects that failed to get an audit in time.

We categorize these vulnerabilities by 4 different threat levels.

THREAT LEVELS

High Risk

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

Medium Risk

Issues on this level are critical to the smart contract's performance, functionality and should be fixed before moving to a live environment.

Low Risk

Issues on this level are minor details and warning that can remain unfixed.

Informational

Information level is to offer suggestions for improvement of efficacy or security for features with a risk free factor.

FOUND THREATS

⚠ Low Risk

No Tax Applied on Very Small Transfer Amounts

The burn tax is calculated using integer division: `taxAmount = value.mul(BURN_TAX_NUMERATOR).div(BURN_TAX_DENOMINATOR);`. If the `value` of a transfer is less than `BURN_TAX_DENOMINATOR` (which is 100), the `taxAmount` will round down to zero.

Impact: Transfers of amounts less than 100 wei (the smallest unit of the token) will not incur any burn tax. While the economic impact of untaxed dust amounts is likely negligible, this means the 1% tax isn't universally applied to every single non-zero value transfer.

- Recommendation:
  - This behavior is typical for integer division in Solidity. It should be documented and understood. Attempting to tax these minute amounts could cost more in gas than the tax itself. The current implementation is generally acceptable.

FOUND THREATS

~~⚠ Low Risk~~

Fixed

Increased Gas Cost for Token Transfers

The overridden _update function, which implements the 1% burn tax, involves several operations beyond a standard ERC20 transfer: an additional balanceOf read (within the require statement for the total amount), arithmetic operations for tax calculation, a _burn operation (which includes an SSTORE and an event), and the emission of an additional TokensBurnedWithTax event.

Impact: This results in higher gas consumption for every token transfer compared to ERC20 tokens without on-transfer tax logic. While a known trade-off for this type of feature, it can negatively affect user experience and transaction affordability, especially during periods of high network congestion or on chains with higher gas fees.

- Recommendation:
  - This is an inherent cost of the feature. Ensure users are aware of the potentially higher gas fees. No direct code change is advised if the feature is intended to operate as is, but benchmarking gas costs would be prudent.

FOUND THREATS

~~⚠ Low Risk~~

Fixed

Absence of a Pause Mechanism for Emergency Situations

The contract does not inherit or implement any pausing functionality (e.g., OpenZeppelin's Pausable utility). A pause mechanism allows designated addresses to temporarily halt most or all contract functions in case a critical vulnerability or an unexpected issue is discovered post-deployment.

Impact: If a severe bug unrelated to admin controls (e.g., a flaw in the ERC20 interaction, or an issue with the tax mechanism that allows unforeseen exploits) is found, there is no way to stop contract activity to prevent further damage or exploitation without redeploying the entire contract.

- Recommendation:
  - Consider incorporating a pausable mechanism. If added, the control over pausing/unpausing would typically reside with a highly secured address or governance protocol. (While pausing introduces a control element, the lack of it is a risk to the contract's stability itself).

FOUND THREATS

Informational

Contract is Not Upgradeable

The smart contract is not implemented using an upgradeability pattern (e.g., proxy patterns like UUPS or Transparent Upgradeable Proxy). This means its logic is immutable once deployed.

Fixed Maximum Supply

The SCORCH token has a hardcoded MAX_SUPPLY of 15,000,000,000 * (10 ** 18), which cannot be altered. The mint function respects this cap.

Burn-on-Transfer Feature

A 1% tax is applied to token transfers (excluding minting operations or direct burns to address(0)). This tax amount is deducted from the sender and burned, reducing the total supply over time.

FOUND THREATS

Informational

Utilization of OpenZeppelin Standards

The contract leverages established OpenZeppelin libraries for core ERC20 functionality (ERC20.sol), context provision (Context.sol), reentrancy protection (ReentrancyGuard.sol), and role-based access (AccessControl.sol). This is a good practice as these libraries are well-audited and battle-tested.

Comprehensive Event Emissions

The contract correctly emits events for significant actions: TokensBurnedWithTax for taxed burns, standard Transfer events for minting, regular transfers, and direct burns. AccessControl (used for MINTER_ROLE) also emits role-related events. These are essential for off-chain applications, UIs, and transaction tracking.

SUMMARY & CONCLUSION

The **SCORCH** token smart contract exhibits a commendable adherence to established blockchain development practices, notably through its proficient use of OpenZeppelin's industry-standard libraries. This approach significantly strengthens the contract's security posture against many common vulnerabilities.

Following an initial review, the team has proactively enhanced the contract's robustness by integrating an emergency **pause mechanism**, providing critical protection for users, and by **removing external libraries** in favor of relying on the native security features of the modern Solidity compiler.

The core tokenomics, including a fixed maximum supply and the innovative 1% burn tax mechanism, remain implemented with clarity. The logic for applying the tax is transparent, and essential events are diligently emitted to foster on-chain transparency. The project's design choices, such as the gas implications of the on-transfer tax or the current approach to contract upgradeability, represent strategic decisions with inherent trade-offs.

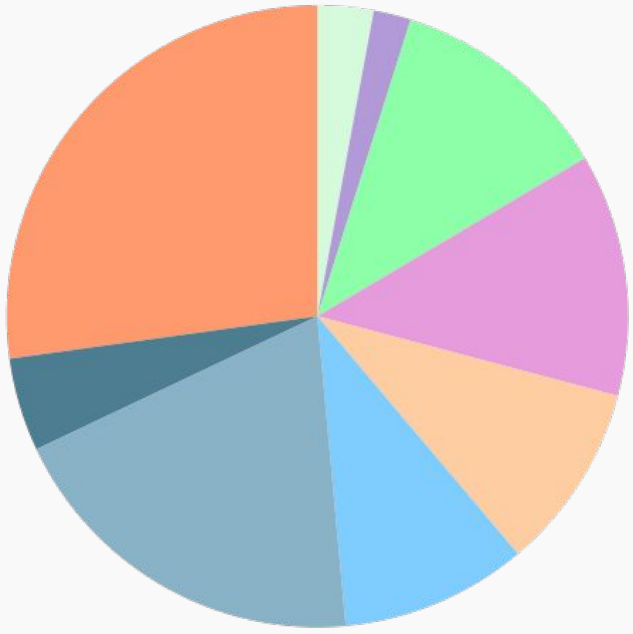
In essence, the SCORCH token contract showcases a well-structured and secure design, further improved by the team's commitment to best practices. As with any smart contract deployment, a final review of the updated code is an indispensable step to ensure the utmost security and readiness for launch.

TOKENOMICS

*The following tokenomics are based on the project's whitepaper and/or website:

Tokens distribution

- Team & Founders (3.33%)
- Advisors & Partnerships (1.67%)
- Ecosystem Development (11.67%)
- Pre-Sale (13.33% = 2 billion tokens)
- Liquidity Pool (10%)
- Community Rewards/Airdrops (10%)
- Staking Pools (20%)
- Expansion Plans (5%)
- Burning Platform Reserve (28.33%)



SPYWOLF

CRYPTO SECURITY

Audits | KYCs | dApps
Contract Development

ABOUT US

We are a growing crypto security agency offering audits, KYCs and consulting services for some of the top names in the crypto industry.

- **OVER 700 SUCCESSFUL CLIENTS**
- **MORE THAN 1000 SCAMS EXPOSED**
- **MILLIONS SAVED IN POTENTIAL FRAUD**
- **PARTNERSHIPS WITH TOP LAUNCHPADS, INFLUENCERS AND CRYPTO PROJECTS**
- **CONSTANTLY BUILDING TOOLS TO HELP INVESTORS DO BETTER RESEARCH**

To hire us, reach out to
contact@spywolf.co or
t.me/joe_SpyWolf

FIND US ONLINE

SPYWOLF.CO

@SPYWOLFNETWORK

@SPYWOLFNETWORK

Disclaimer

This report shows findings based on our limited project analysis, following good industry practice from the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, overall social media and website presence and team transparency details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report.
While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the disclaimer below – please make sure to read it in full.

DISCLAIMER:
By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice.
No one shall have any right to rely on the report or its contents, and SpyWolf and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (SpyWolf) owe no duty of care towards you or any other person, nor does SpyWolf make any warranty or representation to any person on the accuracy or completeness of the report.
The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and SpyWolf hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, SpyWolf hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against SpyWolf, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report. The analysis of the security is purely based on the smart contracts, website, social media and team.
No applications were reviewed for security. No product code has been reviewed.