

SMART CONTRACT SECURITY AUDIT OF WEEKEND COIN



SMART CONTRACT AUDIT | SOLIDITY DEVELOPMENT & TESTING | KYC | PROJECT EVALUATION

RELENTLESSLY SECURING THE PUBLIC BLOCKCHAIN

Summary

Auditing Firm InterFi Network

Client Firm Weekend Coin

Architecture InterFi "Echelon" Auditing Standard

Language Solidity

Mandatory Audit Check Static, Software, Auto Intelligent & Manual Analysis

Final Report Date January 05, 2022

<u>Audit Summary</u>

InterFi team has performed a line-by-line manual analysis and automated review of the smart contract. The smart contract was analyzed mainly for common smart contract vulnerabilities, exploits, and manipulation hacks. According to the smart contract audit:

Weekend Coin's smart contract source code has LOW RISK SEVERITY

For the detailed understanding of risk severity, source code vulnerability, and functional test, kindly refer to the audit.

- Contract address: 0x9BCC77f7B66d032d61389fAb024DEBE84b594a6d
- Blockchain: Binance Smart Chain
- ✓ Verify the authenticity of this report on InterFi's GitHub: https://github.com/interfinetwork



Table Of Contents

Project Information

Overvie	eW	∠
InterFi "Eche	elon" Audit Standard	
Audit Sc	cope & Methodology	6
InterFi's	Risk Classification	8
Smart Contro	act Risk Assessment	
Static A	re Analysis	
Software	re Analysis	10
Manual	l Analysis	1
SWC Att	tacks	12
Risk Sta	itus & Radar Chart	16
Report Sumn	<u>nary</u>	
Auditor'	's Verdict	17
<u>Legal Adviso</u>	<u>ory</u>	
Importa	ant Disclaimer	18
About Ir	nterFi Network	19



Project Overview

InterFi was consulted by Weekend Coin to conduct the smart contract security audit of their solidity source code.

About Weekend Coin

Weekendcoin is the World's Best exchange project backed by big team with high blockchain, Dev skills, IT knowledge, project & community management and marketing experiences. WeekendCoin has been learning lessons from his coin father BTC, ETH, BNB and some other big coins. A new crypto birthed by fans of the blockchain online community. WeekendCoin attempts to impress his fathers by providing new and improved features like Exchange, Pool, Swap, Launchpad, NFTs, and more. WeekendCoin is not hyper-deflationary but we intend to burn the 25 million WKND tokens immediately on launch. and we have plan to reward holders with weekend rewards each weekend as part of WeekendCoin's mission to impress his fathers.

Project	Weekend Cointin Cointi
Blockchain	Binance Smart Chain AUOIL
Language	Solidity
Contract	0x9BCC77f7B66d032d61389fAb024DEBE84b594a6d
Website	https://WeekendCoin.space
Telegram Group	https://t.me/weekendcoinofficial
Twitter	https://twitter.com/weekend_coin
Instagram	https://www.Instagram/weekendcoinofficial
Facebook	https://www.facebook.com/weekendcoinofficial



Project Logo



Solidity Source Code On Blockchain (Verified Contract Source Code)

https://bscscan.com/address/0x9bcc77f7b66d032d61389fab024debe84b594a6d#code

Contract Name: Token

Compiler Version: v0.8.2

Optimization Enabled: No with 200 runs

Solidity Source Code On InterFi GitHub

https://github.com/interfinetwork/audited-codes/blob/main/WeekendCoin.sol

SHA-1 Hash

Solidity source code is audited at hash #165d61ab437684c8ecb189376fb2545d3e86a03e



Audit Scope & Methodology

The scope of this report is to audit the smart contract source code of Weekend Coin. InterFi has scanned the contract and reviewed the project for common vulnerabilities, exploits, hacks, and back-doors. Below is the list of commonly known smart contract vulnerabilities, exploits, and hacks:

Category

- Re-entrancy
- Unhandled Exceptions
- Transaction Order Dependency
- Integer Overflow
- Unrestricted Action
- Incorrect Inheritance Order
- Typographical Errors
- Requirement Violation
- Ownership Takeover
- Gas Limit and Loops
- Deployment Consistency
- Repository Consistency
- Data Consistency
- Token Supply Manipulation
- Access Control and Authorization
- Operations Trail and Event Generation
- Assets Manipulation
- Liquidity Access

Smart Contract Vulnerabilities

Source Code Review

Functional Assessment



InterFi's Echelon Audit Standard

The aim of InterFi's "Echelon" standard is to analyze the smart contract and identify the vulnerabilities and the hacks in the smart contract. Mentioned are the steps used by ECHELON-1 to assess the smart contract:

- 1. Solidity smart contract source code reviewal:
 - Review of the specifications, sources, and instructions provided to InterFi to make sure we understand the size, scope, and functionality of the smart contract.
 - Manual review of code, which is the process of reading source code line-byline to identify potential vulnerabilities.
- 2. Static, Manual, and Software analysis:
 - Test coverage analysis, which is the process of determining whether the test cases are covering the code and how much code is exercised when we run those test cases.
 - Symbolic execution, which is analysing a program to determine what inputs causes each part of a program to execute.
- 3. Best practices review, which is a review of the smart contracts to improve efficiency, effectiveness, clarify, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.
- 4. Specific, itemized, actionable recommendations to help you take steps to secure your smart contracts

Automated 3P frameworks used to assess the smart contract vulnerabilities

- Slither
- Consensys MythX, Mythril
- SWC Registry
- Solidity Coverage
- Open Zeppelin Code Analyzer
- Solidity Code Complier



InterFi's Risk Classification

Smart contracts are generally designed to manipulate and hold funds denominated in ETH/BNB. This makes them very tempting attack targets, as a successful attack may allow the attacker to directly steal funds from the contract. Below are the typical risk levels of a smart contract:

Vulnerable: A contract is vulnerable if it has been flagged by a static analysis tool as such. As we will see later, this means that some contracts may be vulnerable because of a false-positive.

Exploitable: A contract is exploitable if it is vulnerable and the vulnerability could be exploited by an external attacker. For example, if the "vulnerability" flagged by a tool is in a function which requires to own the contract, it would be vulnerable but not exploitable.

Exploited: A contract is exploited if it received a transaction on the main network which triggered one of its vulnerabilities. Therefore, a contract can be vulnerable or even exploitable without having been exploited.

of the second se			
Risk severity	Meaning Security Audit		
	This level vulnerabilities could be exploited easily, and can lead to asset loss,		
! Critical	data loss, asset, or data manipulation. They should be fixed right away.		
	This level vulnerabilities are hard to exploit but very important to fix, they carry		
! High	an elevated risk of smart contract manipulation, which can lead to high-risk		
	severity		
	This level vulnerabilities should be fixed, as they carry an inherent risk of future		
! Medium	exploits, and hacks which may or may not impact the smart contract execution.		
	This level vulnerabilities can be ignored. They are code style violations, and		
! Low	informational statements in the code. They may not affect the smart contract		
	execution		



Smart Contract – Static Analysis

Symbol	Meaning
	Function can be modified
Ø s ā	Function is payable
	Function is locked
	Function can be accessed
!	Important functionality
<pre> **Token** Implementation L <constructor> Public ! </constructor></pre>	

Smart Contract
Security Audit



Smart Contract - Software Analysis

Function Signatures

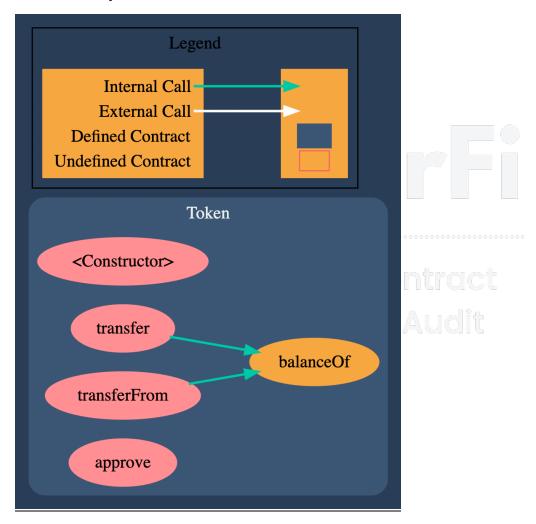
70a08231 => balanceOf(address)

a9059cbb => transfer(address,uint256)

23b872dd => transferFrom(address,address,uint256)

095ea7b3 => approve(address,uint256)

Callout Graph





Smart Contract – Manual Analysis

Function	Description	Tested	Verdict
Total Supply	provides information about the total token	.,	Passed
Total Supply	supply	Yes	
Dalama a Of	provides account balance of the owner's	Yes	Passed
Balance Of	account		
T	executes transfers of a specified number of		Passed
Transfer	tokens to a specified address	Yes	
_	allow a spender to withdraw a set number of		Passed
Approve	tokens from a specified account	Yes	
	returns a set number of tokens from a spender to		NA
Allowance	the owner	NA	
	is an action in which the project buys back its		
Buy Back	tokens from the existing holders usually at a	NA	NA
	market price market Contract		
	executes transfers of a specified number of	NA	NA
Burn	tokens to a burn address		
	executes creation of a specified number of		
Mint	tokens and adds it to the total supply	NA	NA
	circulating token supply adjusts (increases or		
Rebase	decreases) automatically according to a token's	NA	NA
	price fluctuations	•	
	stops specified wallets from interacting with the	NA	NA
Blacklist	smart contract function modules		
	stops or locks all function modules of the smart	NA	
Lock	contract		NA



Function	Description	Tested	Verdict
Dividend	executes transfers of a specified dividend token	NA	NA
	to a specified address		
Airdrop	executes transfers of a specified number of	N.1.6	NA
Allalop	tokens to a specified address	NA	
Max Transaction	a non-whitelisted wallet can only transfer a		NA
Max Transaction	specified number of tokens	NA	
Max Wallet	a non-whitelisted wallet can only hold a		
Max wallet	specified number of tokens	NA	NA
	functionality to limit the number of transactions		
Cooldown Timer	that a wallet can make within a 24-hour span	NA	NA
	stops some or all bot wallets from interacting		
Anti Bot	with the smart contract	NA	NA
	prevents bots from making transaction at		
Anti Snipe	"addLiquidity" block	NA	NA
	executes transfer of contract ownership to a		
Transfer Ownership	specified wallet	NA	NA
Renounce	executes transfer of contract ownership to a		
Ownership	dead address	NA	NA



Best Practices V

- Owner cannot stop or pause the smart contract.
- Owner cannot lock or burn the user assets.
- Owner cannot mint tokens after initial contract creation/deployment.
- Be aware that active smart contract owner privileges constitute an elevated impact to smart contract's safety and security.
- The smart contract has low severity issue which may or may not create any functional vulnerability.

```
"resource": " /WeekendCoin.sol",
"owner": "_generated_diagnostic_collection_name_#0",
"severity": 8, (! Low Severity)

"SPDX license identifier not provided in source file.",
}
```

The smart contract has low severity issue which may or may not create any functional vulnerability.

```
"resource": "/WeekendCoin.sol",

"owner": "_generated_diagnostic_collection_name_#0",

"severity": 8, (! Low Severity)

"Function state mutability can be restricted to view.", [22,5]
}
```



Smart Contract - SWC Attacks

SWC ID	Description	Verdict
SWC-101	Integer Overflow and Underflow	Passed
SWC-102	Outdated Compiler Version	! Low
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	Re-entrancy	Passed
SWC-108	State Variable Default Visibility	Passed
SWC-109	Uninitialized Storage Pointer	Passed
SWC-110	Assert Violation Smart Contract	Passed
swc-111	Use of Deprecated Solidity Functions	Passed
SWC-112	Delegate Call 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



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	! Low
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 (Irrelevant/Dead Code)	Passed
SWC-136	Unencrypted Private Data On-Chain	Passed



Smart Contract - Risk Status & Radar Chart

Risk Severity Status ! Critical None critical severity issues identified None high severity issues identified ! High ! Medium None medium severity issues identified 2 low severity issues identified ! Low Verified 54 functions and instances verified and checked Score out of 100 Compiler Check 85 Interface Safety Static Analysis 80 Manual Analysis Software Analysis



Auditor's Verdict

InterFi team has performed a line-by-line manual analysis and automated review of the smart contract. The smart contract was analyzed mainly for common smart contract vulnerabilities, exploits, and manipulation hacks.

Weekend Coin's smart contract source code has LOW RISK SEVERITY



Note for stakeholders

- Be aware that active smart contract owner privileges constitute an elevated impact on smart contract's safety and security.
- Make sure that the project team's KYC/identity is verified by an independent firm, e.g., InterFi.
- Always check if the contract's liquidity is locked. A longer liquidity lock plays an important role in project's longevity. It is recommended to have multiple liquidity providers.
- Examine the unlocked token supply in the owner, developer, or team's private wallets.
 Understand the project's tokenomics, and make sure the tokens outside of the LP Pair are vested or locked for a longer period of time.
- Ensure that the project's official website is hosted on a trusted platform, and is using an active SSL certificate. The website's domain should be registered for a longer period of time.



Important Disclaimer

InterFi Network provides contract auditing and project verification services for blockchain projects. The purpose of the audit is to analyse the on-chain smart contract source code, and to provide basic overview of the project. This report should not be transmitted, disclosed, referred to, or relied upon by any person for any purposes without InterFi's prior written consent.

InterFi provides the easy-to-understand assessment of the project, and the smart contract (otherwise known as the source code). The audit makes no statements or warranties on the security of the code. It also cannot be considered as an enough assessment regarding the utility and safety of the code, bug-free status, or any other statements of the contract. While we have used all the data at our disposal to provide the transparent analysis, it is important to note that you should not rely on this report only — we recommend proceeding with several independent audits and a public bug bounty program to ensure the security of smart contracts. Be aware that smart contracts deployed on a blockchain aren't resistant from external vulnerability, or a hack. Be aware that active smart contract owner privileges constitute an elevated impact to smart contract's safety and security. Therefore, InterFi does not guarantee the explicit security of the audited smart contract.

The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.

This report should not be considered as an endorsement or disapproval of any project or team.

The information provided on this report does not constitute investment advice, financial advice, trading advice, or any other sort of advice and you should not treat any of the report's content as such. Do conduct your own due diligence and consult your financial advisor before making any investment decisions.



About InterFi Network

InterFi Network provides intelligent blockchain solutions. InterFi is developing an ecosystem that is seamless and responsive. Some of our services: Blockchain Security, Token Launchpad, NFT Marketplace, etc. InterFi's mission is to interconnect multiple services like Blockchain Security, DeFi, Gaming, and Marketplace under one ecosystem that is seamless, multi-chain compatible, scalable, secure, fast, responsive, and easy-to-use.

InterFi is built by a decentralized team of UI experts, contributors, engineers, and enthusiasts from all over the world. Our team currently consists of 6+ core team members, and 10+ casual contributors. InterFi provides manual, static, and automatic smart contract analysis, to ensure that project is checked against known attacks and potential vulnerabilities.

To learn more, visit https://interfi.network

To view our audit portfolio, visit https://github.com/interfinetwork

To book an audit, message https://t.me/interfiaudits



