TrustFacture

SECURITY AUDIT REPORT

WITCH CAT FINANCE

COMPLETED ON:
23.11.2022

OVERVIEW

The purpose of this report is to audit the smart contract source code of Witch Cat Finance (WCF), their website and social media. TrustFacture 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:

Smart Contract Vulnerabilities:

Re-entrancy
Unhandled Exceptions
Transaction Order Dependency
Integer Overflow
Unrestricted Action
Incorrect Inheritance Order
Typographical Errors
Requirement Violation

Source Code Review:

Ownership Takeover
Gas Limit and Loops
DeploymentConsistency
Repository Consistency
Data Consistency
Token Supply Manipulation

Functional Assessment:

Access Control and Authorization
Operations Trail and Event Generation
Assets Manipulation
Liquidity Access

ABOUT PROJECT

Witch Cat Finance is a project for investors that combines five platforms Decentralized exchange, Payment Gateway, APY Staking, NFT Marketplace, Metaverse in the same platform, ready to generate passive income.

Contract Token Name	WITCH CAT FINANCE	
Symbol	WCF	
Contract Adress	0xD75F55c4C3944Db6a481e35ee69c4B4452157c60	
Network	Binance Smart Chain	
Total Supply	100,000,000,000 WCF	
Deployment date	Oct-05-2022	
Language	Solidity	
Status	Not Launched	
Taxes	Buy Tax: 2%	Sell Tax: 2%
Presale Launch	On Pinksale	
Realese date	28.11.2022	

TrustFacture

TOKEN STATS

Liquidity	Not added yet
Burn	15,001,000,000 (15.0010%)
LP Adress	Liquidity not added yet
First transfer date	Oct-05-2022
Creator Wallet	0x72300c0fD98eF72E2b8aF0472C7Cd24eebfD8552
Liquidity Percent	60% of supply
Liquidity Lockup Time	365 days after pool ends
Transfer Count	9

TOKEN HOLDERS

Rank	Address	Quantity	%
1.	Pinksale: PinkLock V2	45,000,000,000	45%
2.	0xf3a2ade790c5ee42272768c16108616a95680003	39,999,000,000	39.999%
3.	Null Address: 0x000...dEaD	15,001,000,000	15.001%

TrustFacture

VULNERABILITY CHECK

CONTRACT FUNCTIONS:

1. function approve(address spender, uint256 amount) external override returns (bool)
2. function approveContractContingency() external onlyOwner returns (bool)
3. function enableTrading() public onlyOwner
4. function excludePresaleAddresses(address router, address presale) external onlyOwner
5. function lockTaxes() external onlyOwner
6. function multiSendTokens(address[] memory accounts, uint256[] memory amounts) external onlyOwner
7. function removeSniper(address account) external onlyOwner
8. function renounceOwnership() external onlyOwner
9. function renounceOriginalDeployer() external
10. function setNewRouter(address newRouter) external onlyOwner
11. function setLpPair(address pair, bool enabled) external onlyOwner
12. function setInitializer(address initializer) external onlyOwner
13. function setExcludedFromLimits(address account, bool enabled) external onlyOwner
14. function setExcludedFromFees(address account, bool enabled) public onlyOwner
15. function setExcludedFromProtection(address account, bool enabled) external onlyOwner
16. function setProtectionSettings(bool _antiSnipe, bool _antiBlock) external onlyOwner
17. function setTaxes(uint16 buyFee, uint16 sellFee, uint16 transferFee) external onlyOwner
18. function setWallets(address payable marketing) external onlyOwner
19. function setSwapSettings(uint256 thresholdPercent, uint256 thresholdDivisor, uint256 amountPercent, uint256 amountDivisor) external onlyOwner
20. function setPriceImpactSwapAmount(uint256 priceImpactSwapPercent) external onlyOwner
21. function setContractSwapEnabled(bool swapEnabled, bool priceImpactSwapEnabled) external onlyOwner
22. function setOperator(address newOperator) public
23. function sweepContingency() external onlyOwner
24. function transferOwner(address newOwner) external onlyOwner
25. function transfer(address recipient, uint256 amount) public override returns (bool)
26. function transferFrom(address sender, address recipient, uint256 amount) external override returns (bool)

CHECKLIST:

No	Description	Results
1.	Compiler errors.	Passed
2.	Possible delays in data delivery.	Passed
3.	Timestamp dependence.	Passed
4.	Integer Overflow and Underflow.	Passed
5.	Race Conditions and Reentrancy.	Passed
6.	DoS with Revert.	Passed
7.	DoS with block gas limit.	Passed
8.	Methods execution permissions.	Passed
9.	Economy model of the contract.	Passed
10.	Private user data leaks.	Passed
11.	Malicious Events Log.	Passed
12.	Scoping and Declarations.	Passed
13.	Uninitialized storage pointers.	Passed
14.	Arithmetic accuracy.	Passed
15.	Design Logic.	Passed
16.	Impact of the exchange rate	Passed
17.	Oracle Calls.	Passed
18.	Cross-function race conditions.	Passed
19.	Fallback function security.	Passed
20.	Safe Open Zeppelin contracts and implementation usage.	Passed

TrustFacture

STATUS:

Main Category	Subcategory	Results
Contract Programming	Solidity version not specified.	Passed
	Solidity version too old.	Passed
	Integer overflow/underflow.	Passed
	Function input parameters lack of check.	Passed
	Function input parameters check bypass.	Passed
	Function access control lacks management.	Passed
	Critical operation lacks event log.	Passed
	Human/contract checks bypass.	Passed
	Random number generation/use vulnerability.	Passed
	Fallback function misuse.	Passed
	Race condition.	Passed
	Logical vulnerability.	Passed
	Other programming issues.	Passed
Code Specification	Visibility not explicitly declared.	Passed
	Var. storage location not explicitly declared.	Passed
	Use keywords/functions to be deprecated.	Passed
	Other code specification issues.	Passed
Gas Optimization	Assert () misuse.	Passed
	High consumption 'for/while' loop.	Passed
	High consumption 'storage' storage.	Passed
	"Out of Gas" Attack.	Passed
Business Risk	The maximum limit for mintage not set.	Passed
	"Short Address" Attack.	Passed
	"Double Spend" Attack.	Passed

CONCLUSION:

The Smart Contract code passed the audit. We have used all possible tests based on given objects as files. Since possible test cases can be unlimited for such extensive smart contract protocol, hence we provide no such guarantee of future outcomes. We have used all the latest static tools and manual observations to cover maximum possible test cases to scan everything. Smart Contracts within the scope were manually reviewed and analyzed with static analysis tools. Smart Contract's high-level description of functionality was presented in Status section of the report. Audit report contains all found security vulnerabilities and other issues in the reviewed code.

Security state of the reviewed contract is "**Well Secured**".

Our Methodology

We like to work with a transparent process and make our reviews a collaborative effort. The goals of our security audits are to improve the quality of systems we review and aim for sufficient remediation to help protect users. The following is the methodology we use in our security audit process.

Manual Code Review:

In manually reviewing all of the code, we look for any potential issues with code logic, error handling, protocol and header parsing, cryptographic errors, and random number generators. We also watch for areas where more defensive programming could reduce the risk of future mistakes and speed up future audits. Although our primary focus is on the in-scope code, we examine dependency code and behavior when it is relevant to a particular line of investigation.

Vulnerability Analysis:

Our audit techniques included manual code analysis, user interface interaction, and whitebox penetration testing. We look at the project's web site to get a high-level understanding of what functionality the software under review provides. We then meet with the developers to gain an appreciation of their vision of the software. We install and use the relevant software, exploring the user interactions and roles. While we do this, we brainstorm threat models and attack surfaces. We read design documentation, review other audit results, search for similar projects, examine source code dependencies, skim open issue tickets, and generally investigate details other than the implementation.

Documenting Results:

We follow a conservative, transparent process for analyzing potential security vulnerabilities and seeing them through successful remediation. Whenever a potential issue is discovered, we immediately create an Issue entry for it in this document, even though we have not yet verified the feasibility and impact of the issue. This process is conservative because we document our suspicions early even if they are later shown to not represent exploitable vulnerabilities. Code analysis is the most tentative, and we strive to provide test code, log captures, or screenshots demonstrating our confirmation. After this we analyze the feasibility of an attack in a live system.

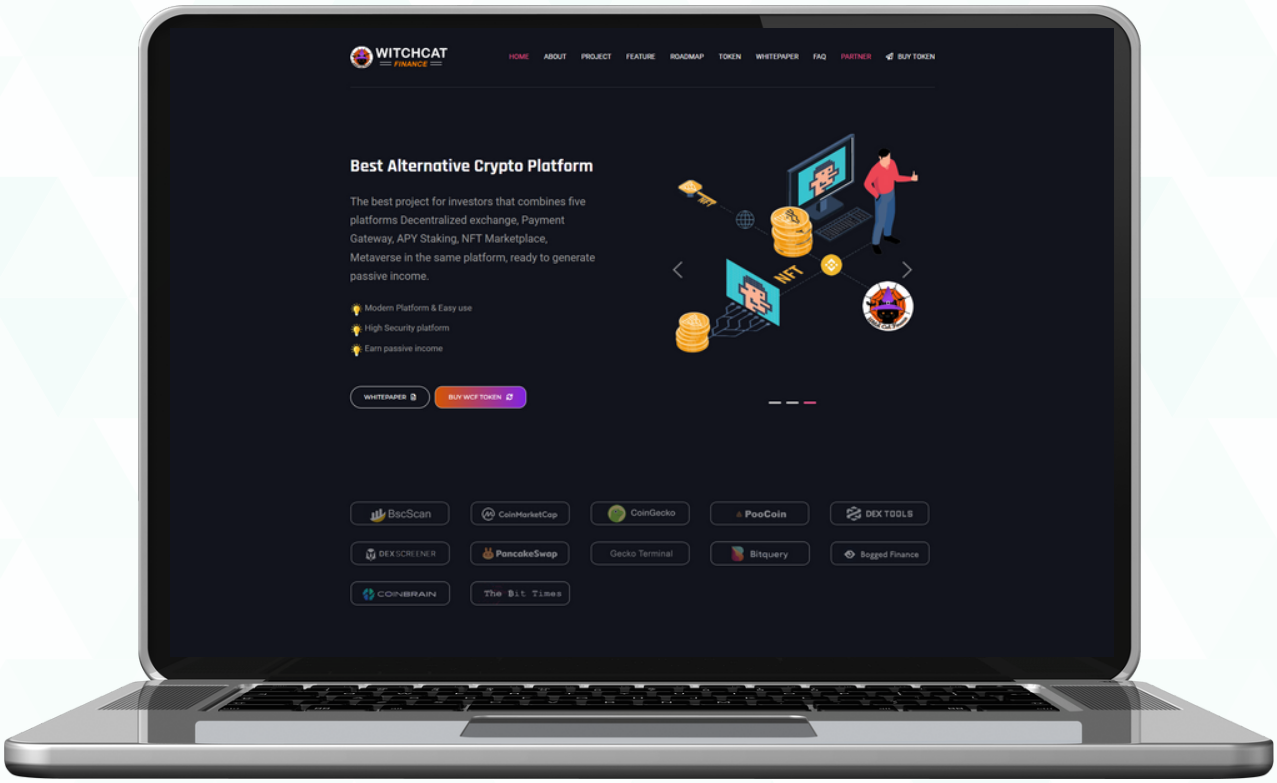
RISK SEVERITY

Risk Severity	Definition
! Cricital	No Cricital severity issues found
! High	No High severity issues found
! Medium	No Medium severity issues found
! Low	No Low severity issues found
Verified	46 Functions and instances checked
Safety Score	100 out of 100

WITCH CAT FINANCE (WCF) has not shown any critical vulnerabilities or errors, the project has a transparent contract with no hidden features.
In the next part of the audit, we will check the website and social media of LITE ROCKET TOKEN project.

TrustFacture

WEBSITE



www.witchcatfinance.io

Google Speed Insight Score:
75/100

Domain Registration Date::
2022-08-23

Security Test::
Passed

Domain Expire Date::
2023-08-23

Whitepaper::
Realesed

Roadmap::
Detailed, presented in 6 phases

Great one-page clear design with all the necessary information.
The site is secure and has an SSL certificate. Optimized for mobile and PC devices. CLS optimalization is great. Only the speed on Mobile devices and optimization of images may need improvement for better results.

SOCIAL MEDIA

The site has active social media such as:

Twitter:	Telegram:	Discord:
twitter.com/witchcatfi	t.me/witchcatfiance	
Created in August 2022	3,4k Group members	No Discord
1899 Followers	Active community	
Logo + Banner	Logo and description	

TrustFacture

DISCLAIMER

By reading this report or any part of it, you agree to the terms of this Disclaimer. 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 TrustFacture.com owns no duty of care towards you or any other person, nor does TrustFacture.com 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, TrustFacture.com hereby excludes all representations, warranties, conditions and other terms, TrustFacture.com hereby excludes all liability and responsibility and neither you nor any other person shall have any claim against TrustFacture.com 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 water in delict, tort (including without limitations negligence), contract, breach of statutory duty, misinterpretation (weather 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. This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intended to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology. Blockchain technology and cryptographic assets present a high level of ongoing risk. Snipe. Finance's positions that each company and individual are responsible for their own due diligence and continuous security. TrustFacture.com goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze. The analysis of the security is purely based on the smart contract, website and social media. No applications were reviewed for security.

TrustFacture

ABOUT US

TrustFacture

WE ARE A FAST GROWING AGENCY OFFERING SMART CONTRACT AUDITS, KYC AND EFFECTIVE MARKETING.

WHAT WE OFFER?

FULL SMART CONTRACT AUDIT - WE WILL THOROUGHLY REVIEW YOUR SMART CONTRACT. MANUALLY LINE BY LINE AS WELL AS WITH AUTOMATED TOOLS.
FAST - WE DELIVER OUR AUDITS FROM 6 TO 48H AFTER WE RECEIVE YOUR MONEY.
AFFORDABLE - WE PROVIDE AUDITS AT AN AFFORDABLE PRICE FOR EVERYONE.
PREMIUM AND CLEAN LOOK - OUR AUDITS HAVE A CLEAR DESIGN AND MOST IMPORTANTLY, THEY ARE EASY TO READ FOR NEW INVESTORS.

OUR SERVICES:

PREMIUM SMART CONTRACT SECURITY AUDIT
KYC - ONLINE IDENTITY VERIFICATION PROCESS OFFERS MAXIMUM SECURITY AND TRUST FOR YOUR INVESTORS
SOCIAL MEDIA MARKETING
SEO AND **REVIEW** OF YOUR PROJECT BY OUR TEAM.

CONNECT WITH US:

WEBSITE: WWW.TRUSTFACTURE.COM
TELEGRAM: T.ME/TRUSTWOJTEK
TWITTER: WWW.TWITTER.COM/TRUSTFACTURE