Stream Smart Business

Smart Contract security and compliance assessments

NFTapp.Finance

streamsmartbusiness.com

24th March 2022

Summary

This audit has resulted from the analysis and testing of the Smart Contract against common or uncommon vectors used in code attacks, analysis from the cyber vulnerabilities possible, and also if the code structure follows the best practices in the industry standards.

The works are developed through some methodologies, among them the revision of all lines of the source code.

Reviews result in ratings that range from critical to passing. A rating is given as critical if it allows potential exploitation of the problem by attackers, if the manifestation of the bug leads to financial loss, or if it disrupts code execution.

After the CRITICAL rating we have the SERIOUS, MEDIUM and LOW ratings, with each indicating the severity level.

We can still classify them in INFORMATIONAL, classifications that are just simple indications about the source code, but that do have a classification of LOW severity.

The APPROVED ones are all those that passed all the analyzes and do not fit in the previous cases.

NOT FOUND indications refer to situations not found in the contract, which also means that they have the status of APPROVED.

Overview

The contract has only one file, **NFTAppFinance.sol**, which has a source code of 1622 lines. We have verified that all functions and variables have been used and commented in accordance with Ethereum Blockchain EPI documentation and standards, OpenZeppelin standard libraries and other technical authorities.

Project name	NFTapp.Finance
source code	https://bscscan.com/address/0xf17e7382f937cd1204a674b87e2aa358cd027bf2#code
Contract address	0xf17e7382f937cd1204a674b87e2aa358cd027bf2
ByteCode for Ethereum Virtual Machine	https://bscscan.com/bytecode-decompiler?a=0xf17e7382f937cd1204a674b87e2aa358cd027bf2
Contract fees	10% buy and 15% sell
Liquidity rates	0% buy and 0% sell
Dividend rates	0% buy and 0% sell
Marketing wallet fees	5% buy and 7.5% sell
Development wallet fees	5% buy and 7.5% sell
Website	NFTapp.finance
Telegram	t.me/NFTappBR
Twitter	twitter.com/NFTappfinance

Project description

NFTapp.Finance is a dapp platform that combines the generation of NFTs, from the creation, trading of NFTs to their publication on social network.

Token description

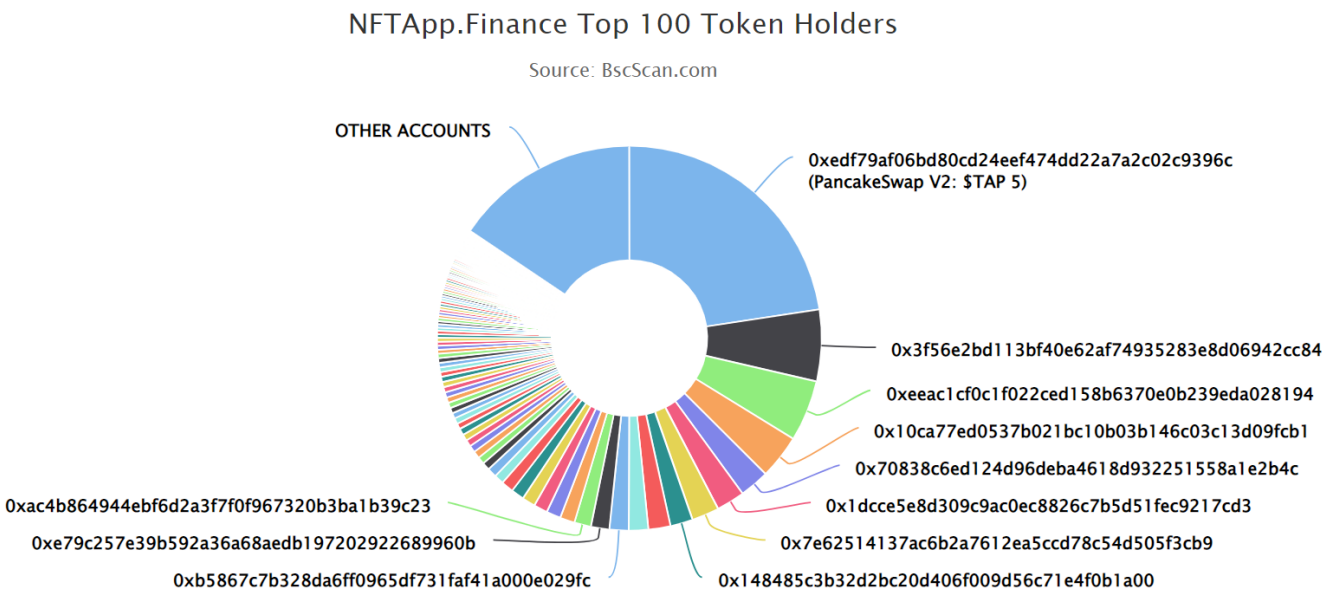
The contract has standard ERC-20 functions, in addition to paying fees for the project's development and marketing portfolios, 10% on purchase and 15% on sale. It also has a (unused) function that makes dividend payments in BNB BSC to all token holders.

There is no accumulation for carrying out LP purchases (known as liquidity injection or liquidity pool fees) in the PancakeSwap liquidity contract.

NFTapp.Finance token distribution

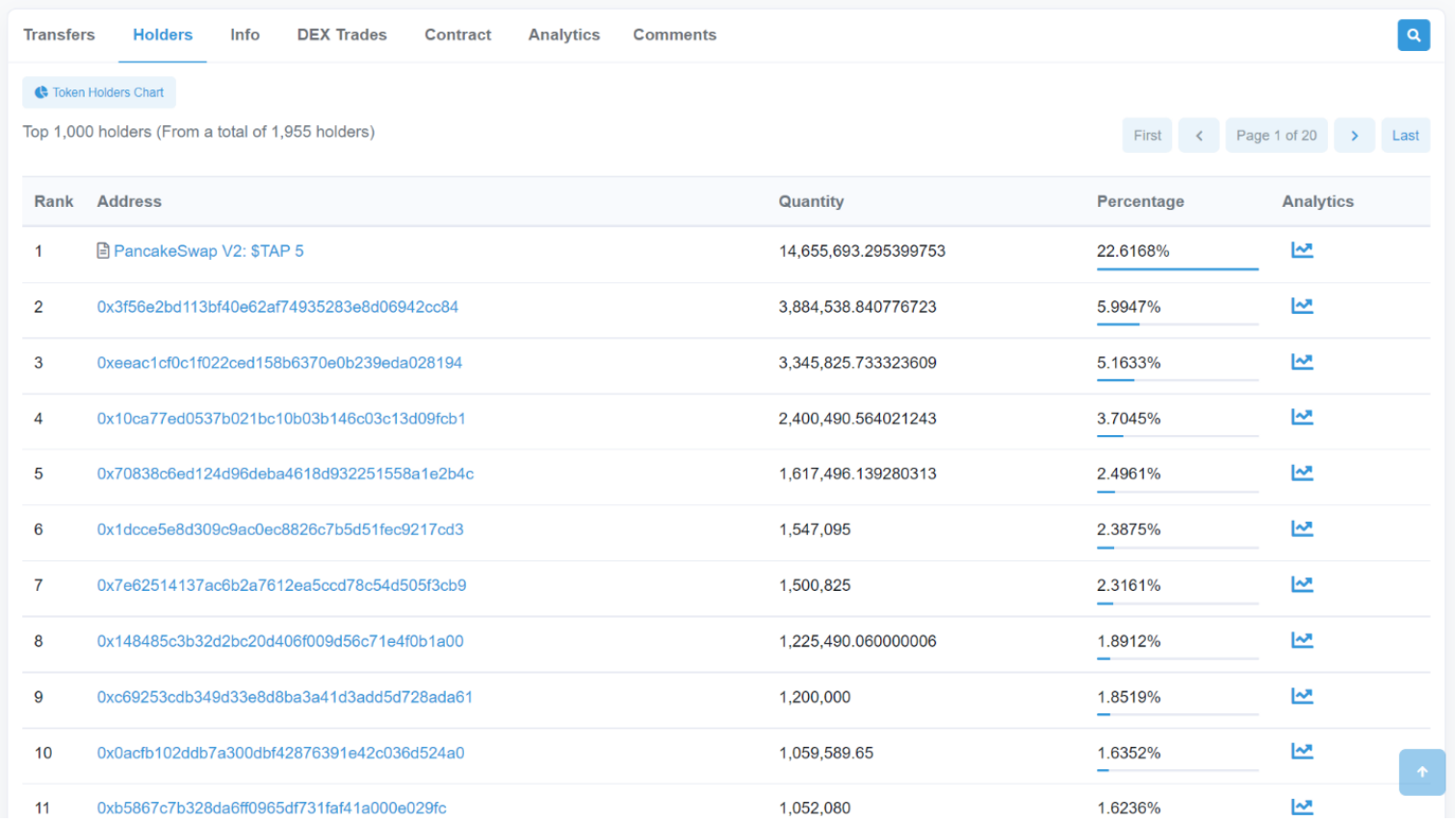
The top 100 holders collectively own 84.44% (54,719,653.73 Tokens) of NFTApp.Finance

Token Total Supply: 64,800,000.00 Token | Total Token Holders: 1,955



(A total of 54,719,653.73 tokens held by the top 100 accounts from the total supply of 64,800,000.00 token)

NFTapp.Finance top 10 token holders



Transfers | Holders | Info | DEX Trades | Contract | Analytics | Comments

Token Holders Chart

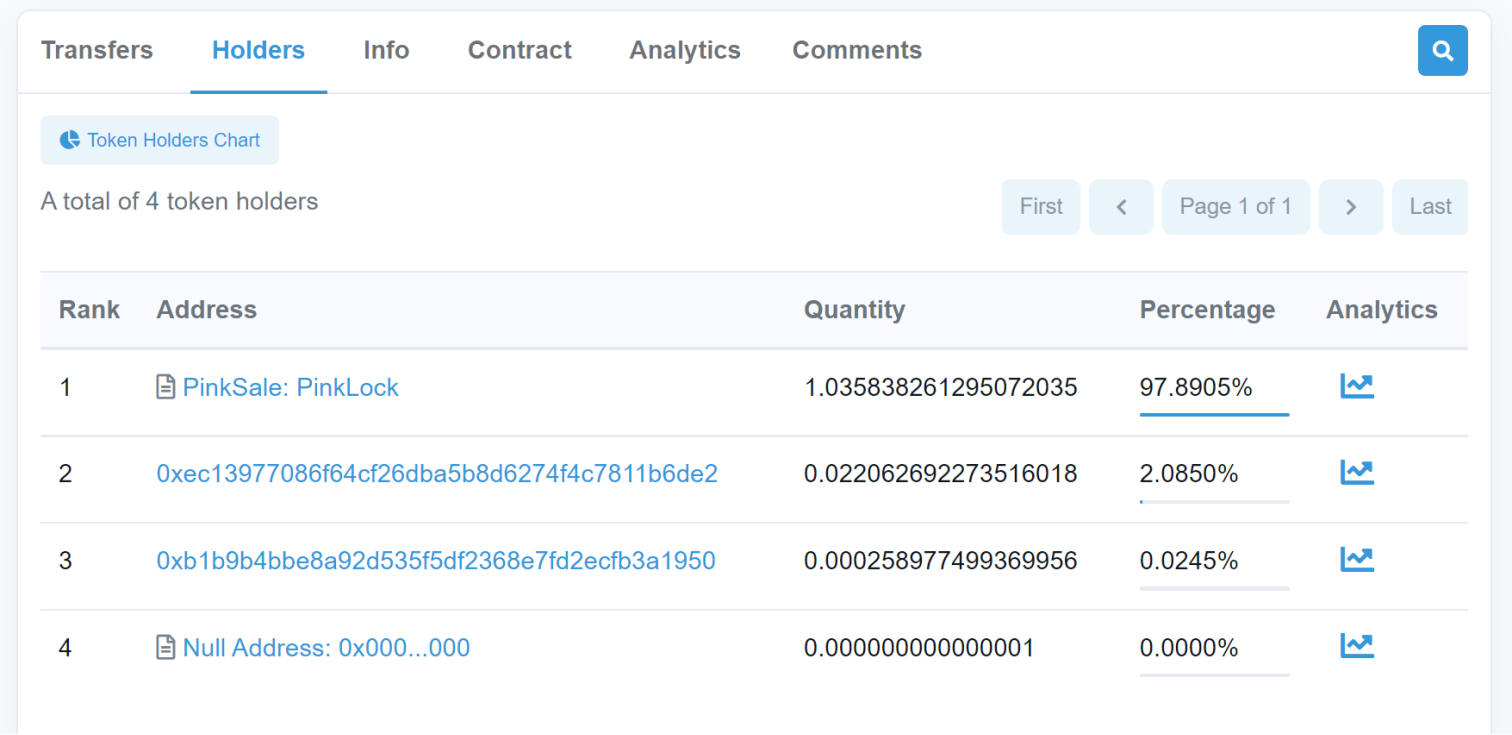
Top 1,000 holders (From a total of 1,955 holders)

First | Page 1 of 20 | Last

Rank	Address	Quantity	Percentage	Analytics
1	PancakeSwap V2: $TAP 5	14,655,693.295399753	22.6168%	
2	0x3f56e2bd113bf40e62af74935283e8d06942cc84	3,884,538.840776723	5.9947%	
3	0xeeac1cf0c1f022ced158b6370e0b239eda028194	3,345,825.733323609	5.1633%	
4	0x10ca77ed0537b021bc10b03b146c03c13d09fcb1	2,400,490.564021243	3.7045%	
5	0x70838c6ed124d96deba4618d932251558a1e2b4c	1,617,496.139280313	2.4961%	
6	0x1dcce5e8d309c9ac0ec8826c7b5d51fec9217cd3	1,547,095	2.3875%	
7	0x7e62514137ac6b2a7612ea5ccd78c54d505f3cb9	1,500,825	2.3161%	
8	0x148485c3b32d2bc20d406f009d56c71e4f0b1a00	1,225,490.060000006	1.8912%	
9	0xc69253cdb349d33e8d8ba3a41d3add5d728ada61	1,200,000	1.8519%	
10	0x0acfb102ddb7a300dbf42876391e42c036d524a0	1,059,589.65	1.6352%	
11	0xb5867c7b328da6ff0965df731faf41a000e029fc	1,052,080	1.6236%	

NFTapp.Finance LP token holders

NFTapp.Finance liquidity holders share an amount of 1,058159 units of LP tokens, with 97.89% of the liquidity blocked and inaccessible for a period of 1 year from the launch of the contract. The rest of the liquidity was added by investors.



Transfers | Holders | Info | Contract | Analytics | Comments

Token Holders Chart

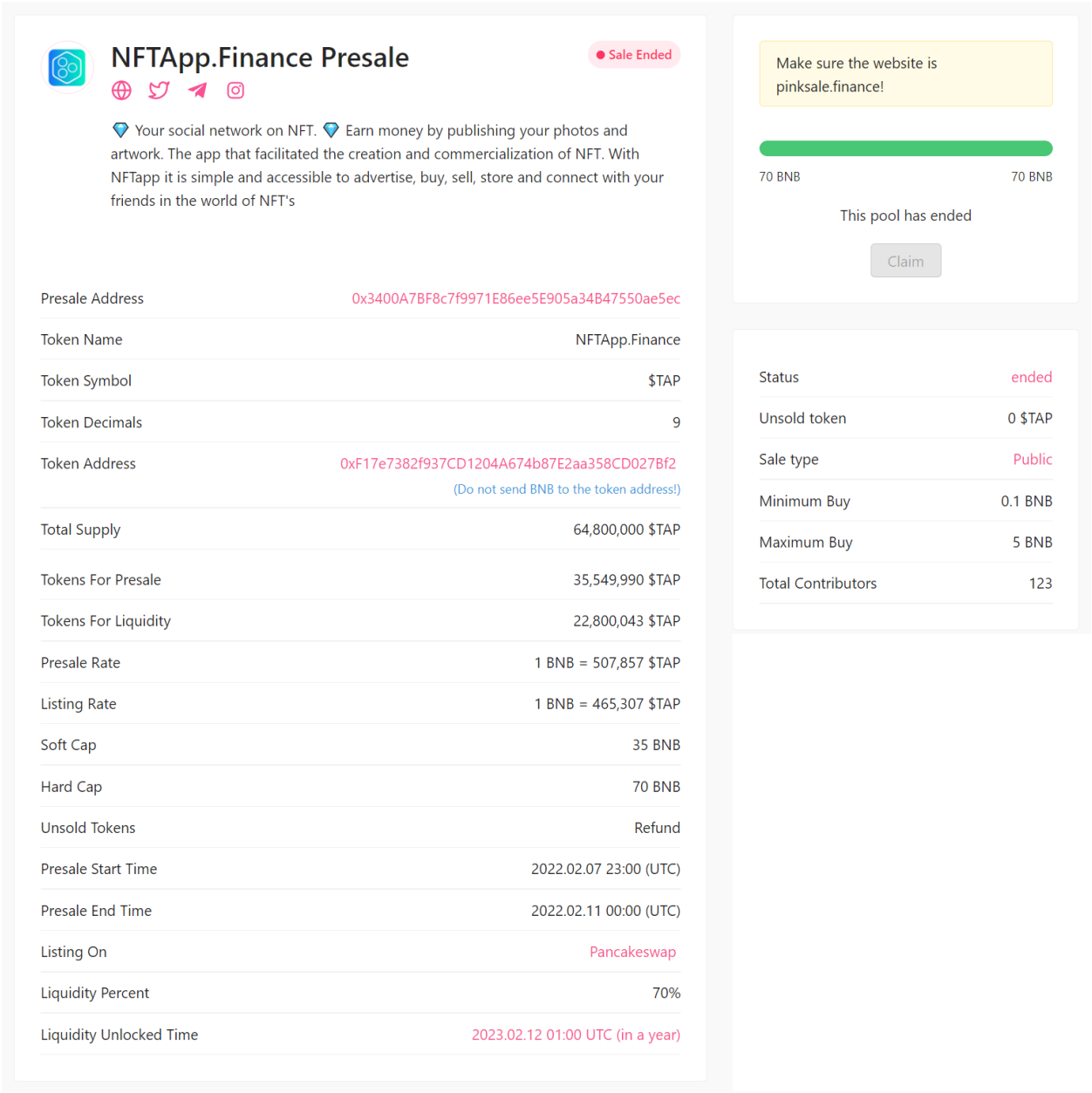
A total of 4 token holders

First | < | Page 1 of 1 | > | Last

Rank	Address	Quantity	Percentage	Analytics
1	PinkSale: PinkLock	1.035838261295072035	97.8905%	
2	0xec13977086f64cf26dba5b8d6274f4c7811b6de2	0.022062692273516018	2.0850%	
3	0xb1b9b4bbe8a92d535f5df2368e7fd2ecfb3a1950	0.000258977499369956	0.0245%	
4	Null Address: 0x000...000	0.000000000000001	0.0000%	

Liquidity lock and presale

Liquidity is blocked for a period of 1 year by PinkSale, the liquidity was blocked automatically after pre-sale.



NFTApp.Finance Presale

Sale Ended

💎 Your social network on NFT. 💎 Earn money by publishing your photos and artwork. The app that facilitated the creation and commercialization of NFT. With NFTapp it is simple and accessible to advertise, buy, sell, store and connect with your friends in the world of NFT's

Presale Address	0x3400A7BF8c7f9971E86ee5E905a34B47550ae5ec
Token Name	NFTApp.Finance
Token Symbol	$TAP
Token Decimals	9
Token Address	0xF17e7382f937CD1204A674b87E2aa358CD027Bf2 (Do not send BNB to the token address!)
Total Supply	64,800,000 $TAP
Tokens For Presale	35,549,990 $TAP
Tokens For Liquidity	22,800,043 $TAP
Presale Rate	1 BNB = 507,857 $TAP
Listing Rate	1 BNB = 465,307 $TAP
Soft Cap	35 BNB
Hard Cap	70 BNB
Unsold Tokens	Refund
Presale Start Time	2022.02.07 23:00 (UTC)
Presale End Time	2022.02.11 00:00 (UTC)
Listing On	Pancakeswap
Liquidity Percent	70%
Liquidity Unlocked Time	2023.02.12 01:00 UTC (in a year)

Make sure the website is pinksale.finance!

70 BNB 70 BNB

This pool has ended

Claim

Status	ended
Unsold token	0 $TAP
Sale type	Public
Minimum Buy	0.1 BNB
Maximum Buy	5 BNB
Total Contributors	123

Checking status of the contract

Code accuracy as per expectations	**APPROVED**
Errors when compiling	**APPROVED**
Timestamp dependency	**APPROVED**
Code testability	**APPROVED**
Clean/lean code	**APPROVED**
DDoS attack per block gas limit	**APPROVED**
DDoS attack by to failed function calls	**APPROVED**
DDoS attack by reversing transaction processing	**APPROVED**
Use of outdated functions in Solidity	**NOT FOUND** / **APPROVED**
Re-entry attack	**APPROVED**
Poorly documented or unprotected SELFDESTRUCT instructions	**NOT FOUND** / **APPROVED**
Not checking return value in Get functions	**APPROVED**
Integer overflow/underflow	**LOW**
Presence of unused variables	**APPROVED**
Code redundancy	**APPROVED**
Gas optimization in transactions	**INFORMATIONAL**
Optimization of contract execution	**APPROVED**
Prevention of loss of values within the contract	**APPROVED**
Require fuction violation	**APPROVED**
Typographical errors	**APPROVED**
Unsecured withdrawal of BNB	**APPROVED**
Logical design	**APPROVED**
Arithmetic precision	**APPROVED**
Security of fallback functions	**APPROVED**
Compliance with EIP standards	**APPROVED**
Property waiver	**LOW**
Implementation of standard and secure OpenZeppelin libraries	**APPROVED**

Privileged fuctions

The contract has functions that can only be modified by the contract owner, which is defined by the ***onlyOwner*** modifier.

The privileged functions define exemption from fees, limits and dividends:

- excludeFromFees (storage in address type 20 bytes and in Boolean logical type 1 or zero), defines the exemption of fees in the transaction;
- excludeFromDividens (storage in address type 20 bytes), defines the exemption from receiving dividends;
- includeInDividends (storage in address type 20 bytes), defines the inclusion in the dividend receiving list;
- setMaxWallet (storage in unsigned integer type, uint256), defines the maximum limit of tokens that each and every wallet can have;
- excludeFormMaxTransaction (storage in address type 20 bytes and in Boolean logical type 1 or zero), defines whether or not a wallet is excluded from the maxWallet limits;
- updateSellPenalty (storage in unsigned integer type, uint256), defines the selling rate increase factor;

The owner also has other privileges in some functions, such as *modifying the gas values (network rate)* and *configuring the conditions of dividend distributions*, if activated.

Rating: INFORMATIONAL

Property waiver

We have verified that the source code does not have any definitions for a disclaimer to deprive the owner of privileged controls over the contract.

Rating: LOW

distributedividends function makes division not indicated

The ***distributeDividends*** function in the dividend contract doesn't make use of safe calculation libraries when it should.

```
function distributeDividends() public override payable {
    require(totalBalance > 0);

      if (msg.value > 0) {
        magnifiedDividendPerShare = magnifiedDividendPerShare.add(
          (msg.value).mul(magnitude) / totalBalance
        );
        emit DividendsDistributed(msg.sender, msg.value);

        totalDividendsDistributed = totalDividendsDistributed.add(msg.value);
      }
}
```

The code snippet is correctly corrected as follows, making use of SafeMath's div function:

```
          (msg.value).mul(magnitude).div(totalBalance)
```

Rating: **LOW**

Overflow and underflow attack

The previous case of unreported division can EVENTUALLY be exploited in overflow or underflow attacks, which would result in execution failures due to errors out of memory space attributed to the storage of the ***magnifiedDividendPerShare*** parameter.

Rating: **LOW**

Rug-pull fuctions

There are no functions or mechanisms in the contract that can be considered rug-pull, such as:

- deliberate token creation schemes (***mint*** function)
- contract self-destruct mechanisms (***selfdestruct*** or ***suicide*** function) to make transactions more difficult to track

The trade starts off at launch of liquidity and must be turned on to allow trading, but it can never be turned off:

```
bool public tradingActive = false;
```

```
// once enabled, can never be turned off
function enableTrading() external onlyOwner {
    tradingActive = true;
    swapEnabled = true;
}
```

Rating: **APPROVED**

Use of indicated standards

The subcontracts used (ERC20 standard, interface, PCSV2 router, LP PancakeV2, dividends) comply with the technical recommendations.

Rating: **APPROVED**

Conclusion

The audit performed manual line-by-line analysis and automated smart contract review. The contract and source code were analyzed primarily from the point of view of common smart contract vulnerabilities, exploits, manipulation hacks, optimization and code structure.

The contract and source code have an **APPROVED** status in the audit reviews and discussions.

No issues were found that fall into CRITICAL, SERIOUS, or MEDIUM status.

Disclaimer

Stream Smart Business and its technical team provide through this audit report only project and contract discussion and evaluation, with the sole intention of helping to improve code quality and improve the level of variation used in smart contract technologies. Therefore, this audit report is not a legally binding document.

This technical audit report is not a buy or sell recommendation, approval or disapproval, or even a definitive answer on the points discussed here, but only expresses the technical and analytical opinion about the **NFTapp.Finance** project.

Reproduction of this material is free and there is no need to ask the creators for permission. However, citation of the source is required.