SPYWOLF

Security Audit Report
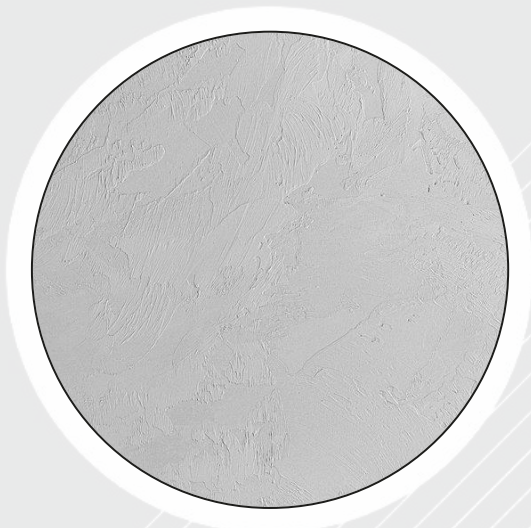
Audit prepared for

Gelato

Completed on

October 22, 2024

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

Gelato

PROJECT DESCRIPTION:

No info yet.

Release Date: Unavailable

Launchpad: Unavailable

Category: Dividend token

01

KEY RESULTS

Cannot mint new tokens	**PASSED**
Cannot pause trading (honeypot)	**PASSED**
Cannot blacklist an address	**PASSED**
Cannot raise taxes over 25%?	**PASSED**
No proxy contract detected	**PASSED**
Not required to enable trading	**PASSED**
No hidden ownership	**PASSED**
Cannot change the router	**PASSED**
No cooldown feature found	**PASSED**
Bot protection delay is lower than 5 blocks	**PASSED** (2 blocks)
Cannot set max tx amount below 0.05% of total supply	**PASSED**
The contract cannot be self-destructed by owner	**PASSED**

For a more detailed and thorough examination of the heightened risks, refer to the subsequent parts of the report.

N/A = Not applicable for this type of contract

*Only new deposits/reinvestments can be paused

CONTRACT INFO

Token Name
Gelato

Symbol
GEL

Contract Address
unavailable

Network
unavailable

Language
Solidity

Deployment Date
unavailable

Contract Type
Dividend token

Total Supply
8,888,888,888

Decimals
18

TAXES

Buy Tax
4%

Sell Tax
4%

*Taxes can be changed in future

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

02

SMART CONTRACT STATS

Calls Count	unavailable
External calls	unavailable
Internal calls	unavailable
Transactions count	unavailable
Last transaction time	unavailable
Deployment Date	unavailable
Create TX	unavailable
Owner	unavailable
Deployer	unavailable

TOKEN TRANSFERS STATS

Transfer Count	unavailable
Total Amount	unavailable
Median Transfer Amount	unavailable
Average Transfer Amount	unavailable
First transfer date	unavailable
Last transfer date	unavailable
Days token transferred	unavailable

03

FEATURED WALLETS

Owner address	Contract is not deployed yet
Marketing fee receiver	Contract is not deployed yet
LP address	Contract is not deployed yet

TOP 3 UNLOCKED WALLETS

unavailable	
unavailable	
unavailable	

04

SPYWOLF.CO

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

05-A

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

05-B

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.

Code Score: 90%

FOUND THREATS

⚠️ **High Risk: 0**

No high risk-level threats found in this contract.

⚠️ **Medium Risk: 0**

No medium risk-level threats found in this contract.

⚠️ **Low Risk: 1**

1 low risk-level threats found in this contract.

08-A

FOUND THREATS

⚠️ Low Risk

Transaction will revert when msg.value is higher than 0, as overflow will occur. plsBefore will be higher than address(this).balance after addLiquidityETH() execution.

```
function addLiquidity(
    uint256 gelAmount,
    uint256 deadline
) public payable nonReentrant {
    _transferFrom(msg.sender, address(this), gelAmount);

    // Track starting balances
    uint256 plsBefore = address(this).balance;
    uint256 gelBefore = balanceOf(address(this));

    pulseRouterV2.addLiquidityETH{value: msg.value}(
        address(this),
        gelAmount,
        0,
        msg.value,
        msg.sender,
        deadline
    );

    // Calculate after balances
    uint256 plsRemaining = address(this).balance.sub(plsBefore);
    uint256 gelRemaining = balanceOf(address(this)).sub(gelBefore);

    if (plsRemaining > 0) {
        (bool success, ) = payable(msg.sender).call{
            value: plsRemaining,
            gas: 30000
        }("");
        require(success, "PLS transfer failed");
    }

    if (gelRemaining > 0) {
        _transferFrom(address(this), msg.sender, gelRemaining);
    }
}
```

- Recommendation:
  - Do not subtract higher number from lower number

08-B

SPYWOLF.CO

ℹ️ Informational: 10

Owner can change contract's auto swap settings.
When auto swap is enabled and swapThreshold is set to 0, swap will fail.
This will not revert the entire transaction as the autoswap occurs in try catch block. It is recommended that swapThreshold's value be always above 1 token.

```solidity
function setSwapBackSettings(
    bool _enabled,
    uint256 _amount
) external authorizedFor(Permission.AdjustContractVariables) {
    swapEnabled = _enabled;
    swapThreshold = _amount;
}

function swapBack() internal swapping {
...........................
uint256 amountGelSwap = swapThreshold.sub(amountGelLiquidity).sub(
        amountGelBurn
    );

    uint256 amountGelSwap = swapThreshold.sub(amountGelLiquidity).sub(
        amountGelBurn
    );

    uint256 balanceBefore = address(this).balance;

    address[] memory path = new address[](2);
    path[0] = address(this);
    path[1] = address(WPLS);

    try
        pulseRouterV2.swapExactTokensForETHSupportingFeeOnTransferTokens(
            amountGelSwap,
            0,
            path,
            address(this),
            block.timestamp
        )
    {
...........................
}
```

08-C

FOUND THREATS

ℹ️ Informational: 10

The contract utilizes roles based access control.
Owner can authorize addresses for contract's authorized settings
functions access, like change fees, exempt from fees etc.

```solidity
function authorizeFor(
    address adr,
    string memory permissionName
) public authorizedFor(Permission.Authorize) {
    uint256 permIndex = permissionNameToIndex[permissionName];
    authorizations[adr][permIndex] = true;
    emit AuthorizedFor(adr, permissionName, permIndex);
}

function authorizeForMultiplePermissions(
    address adr,
    string[] calldata permissionNames
) public authorizedFor(Permission.Authorize) {
    for (uint256 i; i < permissionNames.length; i++) {
        uint256 permIndex = permissionNameToIndex[permissionNames[i]];
        authorizations[adr][permIndex] = true;
        emit AuthorizedFor(adr, permissionNames[i], permIndex);
    }
}

modifier authorizedFor(Permission permission) {
    require(
        !lockedPermissions[uint256(permission)].isLocked,
        "Permission is locked."
    );
    require(
        isAuthorizedFor(msg.sender, permission),
        string(
            abi.encodePacked(
                "Not authorized. You need the permission ",
                permissionIndexToName[uint256(permission)]
            )
        )
    );
    _;
}
```

FOUND THREATS

ℹ️ Informational: 10

Owner can change dividend's distributor distribution criteria like periods between each dividend distribution and amounts of each dividend distribution.

```
function setDistributionCriteria(
    uint256 _minSolidXPeriod,
    uint256 _minSolidXDistribution,
    uint256 _minHexPeriod,
    uint256 _minHexDistribution
) external authorizedFor(Permission.AdjustContractVariables) {
    distributor.setDistributionCriteria(
        _minSolidXPeriod,
        _minSolidXDistribution,
        _minHexPeriod,
        _minHexDistribution
    );
}


function setDistributionCriteria(
    uint256 _minSolidXPeriod,
    uint256 _minSolidXDistribution,
    uint256 _minHexPeriod,
    uint256 _minHexDistribution
) external override onlyToken {
    minSolidXPeriod = _minSolidXPeriod;
    minSolidXDistribution = _minSolidXDistribution;
    minHexPeriod = _minHexPeriod;
    minHexDistribution = _minHexDistribution;
}
```

08-E

FOUND THREATS

ℹ️ Informational: 10

Owner can set fees buy/sell fees up to 10%.
When transfer fees are enabled, transfers will be taxed with the buy
fees' value.
Combined buy+sell = 20%.

```solidity
function setFees(
    uint256 _solidXBurnFee,
    uint256 _stackedBurnFee,
    uint256 _gelatoBurnFee,
    uint256 _solidXReflectionFee,
    uint256 _hexReflectionFee,
    uint256 _liquidityFee,
    uint256 _totalBuyFee,
    uint256 _totalSellFee,
    bool _feesOnNormalTransfers
) external authorizedFor(Permission.AdjustContractVariables) {
    solidXBurnFee = _solidXBurnFee;
    stackedBurnFee = _stackedBurnFee;
    gelatoBurnFee = _gelatoBurnFee;
    solidXReflectionFee = _solidXReflectionFee;
    hexReflectionFee = _hexReflectionFee;
    liquidityFee = _liquidityFee;
    totalBuyFee = _totalBuyFee;
    totalSellFee = _totalSellFee;
    require(
        solidXBurnFee
            .add(stackedBurnFee)
            .add(gelatoBurnFee)
            .add(solidXReflectionFee)
            .add(hexReflectionFee)
            .add(liquidityFee) <= feeDenominator / 10,
        "The total of all combined fees must be 1000 for 100 percent."
    );
    require(totalBuyFee <= feeDenominator / 10, "Buy fee too high");
    require(totalSellFee <= feeDenominator / 10, "Sell fee too high");

    feesOnNormalTransfers = _feesOnNormalTransfers;
}
```

FOUND THREATS

ℹ️ Informational: 10

For the first 2 blocks after token's launch, 99.9% buy/sell fee is applied (automated anti bot measures).
3 blocks after token launch fees are normalized automatically within the current regular fees range (up to 10% buy/sell).

```solidity
function _transferFrom(address sender, address recipient, uint256 amount
) internal returns (bool) {
    ...........................
    uint256 amountReceived = shouldTakeFee(sender, recipient)
    ? takeFee(sender, recipient, amount)
    : amount;
    _balances[recipient] = _balances[recipient].add(amountReceived);
    .........................
}

function getTotalFee(bool selling) public view returns (uint256) {
    if (launchedAt + 1 >= block.number) {
        return feeDenominator.sub(1);
    }
    return selling ? totalSellFee : totalBuyFee;
}

function takeFee(
    address sender,
    address recipient,
    uint256 amount
) internal returns (uint256) {
    uint256 feeAmount = amount.mul(getTotalFee(isSell(recipient))).div(
        feeDenominator
    );

    _balances[address(this)] = _balances[address(this)].add(feeAmount);
    emit Transfer(sender, address(this), feeAmount);

    return amount.sub(feeAmount);
}

function isSell(address recipient) internal view returns (bool) {
    address[] memory liqPairs = pairs;
    for (uint256 i = 0; i < liqPairs.length; i++) {
        if (recipient == liqPairs[i]) return true;
    }
    return false;
}
```

08-G

SPYWOLF.CO

FOUND THREATS

ℹ️ Informational: 10

Owner can exclude address from dividends.

```
function setIsDividendExempt(
    address holder,
    bool exempt
) external authorizedFor(Permission.ExcludeInclude) {
    require(holder != address(this) && holder != pulseV2Pair);
    isDividendExempt[holder] = exempt;
    if (exempt) {
        distributor.setShare(holder, 0);
    } else {
        distributor.setShare(holder, _balances[holder]);
    }
}
```

Owner can set max transaction limit but cannot lower it than 0.2% of total supply.

```
function setTxLimit(
    uint256 amount
) external authorizedFor(Permission.AdjustContractVariables) {
    require(amount >= _totalSupply / 2000);
    _maxTxAmount = amount;
}

function checkTxLimit(address sender, uint256 amount) internal view {
    require(
        amount <= _maxTxAmount || isTxLimitExempt[sender],
        "TX Limit Exceeded"
    );
}
```

08-H

FOUND THREATS

ℹ️ Informational: 10

Owner can add and remove address from liquidity pairs list.

```solidity
function addPair(
    address pair
) external authorizedFor(Permission.AdjustContractVariables) {
    pairs.push(pair);
}

function removeLastPair()
    external
    authorizedFor(Permission.AdjustContractVariables)
{
    pairs.pop();
}
```

Owner can change autoswap's liquidity receiver.

```solidity
function setLiquidityFeeReceiver(
    address _autoLiquidityReceiver
) external authorizedFor(Permission.AdjustContractVariables) {
    autoLiquidityReceiver = _autoLiquidityReceiver;
}
```

Owner can exclude address from fees and max transaction limits.

```solidity
function setIsFeeExempt(
    address holder,
    bool exempt
) external authorizedFor(Permission.ExcludeInclude) {
    isFeeExempt[holder] = exempt;
}

function setIsTxLimitExempt(
    address holder,
    bool exempt
) external authorizedFor(Permission.ExcludeInclude) {
    isTxLimitExempt[holder] = exempt;
}
```

08-I

No information provided about tokenomics.

TOKENOMICS

Website URL:
unavailable

Domain Registry
unavailable

Domain Expiration
unavailable

Technical SEO Test
unavailable

Security Test
unavailable

Design
unavailable

Content
unavailable

Whitepaper
unavailable

Roadmap
unavailable
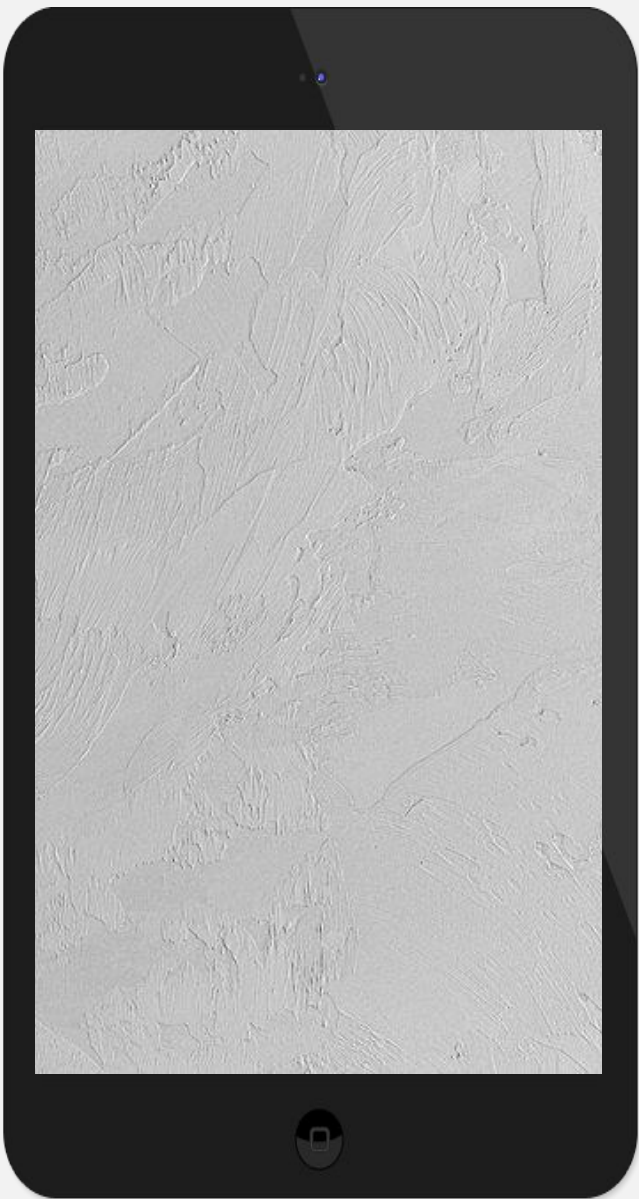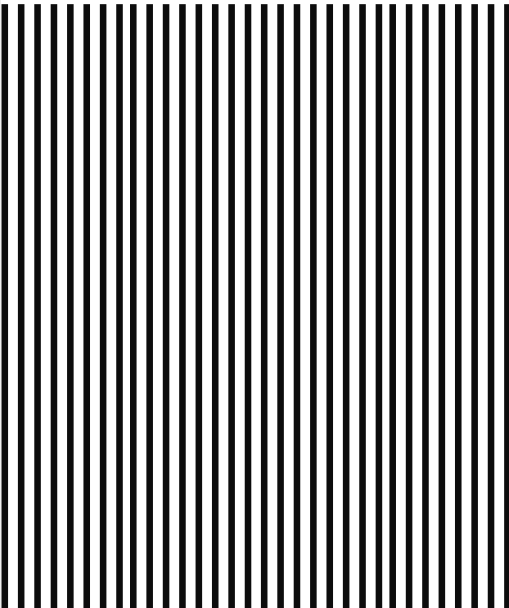
Mobile-friendly?
unavailable

Website Score: 0%

SOCIAL MEDIA

Social Score: 0%

Twitter:
unavailable

Discord
unavailable

Telegram:
unavailable

Medium
unavailable

11

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**

12

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.