PotatoSwap
Audit Report

Fri Oct 17 2025

contact@bitslab.xyz https://twitter.com/scalebit_

ScaleBit

PotatoSwap Audit Report

1 Executive Summary

1.1 Project Information

Description	PotatoSwap is a dex project.
Type	DeFi
Auditors	Bear Two, s3cunda
Timeline	Wed Oct 08 2025 - Fri Oct 10 2025
Languages	Solidity
Platform	Others
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/PotatoSwapFinance/potato_v3_contract
Commits	684396e3c81bf4fa009f586e0544f3fa5f3e2c83

1.2 Files in Scope

The following are the SHA1 hashes of the original reviewed files.

ID	File	SHA-1 Hash
MU3	contracts/help/Multicall3.sol	9add5c4f468ea6bdbfa899d50a2791075a5d7efb
MU2	contracts/help/Multicall2.sol	e4b5c20a697dbf48a20cbe14b6a9b4f1f54e74a3
HV3R	contracts/swapV3periphery/HyperindexV3Router.sol	4617d04a93df2c6d029fddd55dba1dc08291d760
LAM	contracts/swapV3periphery/libraries/LiquidityAmounts.sol	5fb0c1c0783efda5036cf272e83e6fcfce0420bc
PAT	contracts/swapV3periphery/libraries/Path.sol	ab8eac6675add2f1288e9e85ab123abc991fefb3
CID	contracts/swapV3periphery/libraries/ChainId.sol	05e03873fced5e153f5a7888f584b805ac770ada
THE	contracts/swapV3periphery/libraries/TransferHelper.sol	ab970d504bed24fa6d35de2956d3390b98fe9448
CVA	contracts/swapV3periphery/libraries/CallbackValidation.sol	ad2b0f0c95f6bc92e43f6a96c66835c668c42ad1
PTC	contracts/swapV3periphery/libraries/PoolTicksCounter.sol	313d018e8ab86cdddc46e04ed5fef0d75ca48968
BLI	contracts/swapV3periphery/libraries/BytesLib.sol	05794c9a422e19f74b3b0df793daac1e516175c5

PKE	contracts/swapV3periphery/libraries/PositionKey.sol	ab03e197c2f11c85a8159d0a3f0f36ccab30b414
PAD	contracts/swapV3periphery/libraries/PoolAddress.sol	7cea28c4282d1f4fc055f577cbbe05c2d9011f96
SIM	contracts/swapV3periphery/lens/SwapInterfaceMulticall.sol	ea0ff3e4bfda7a9419b27139f9cf150eea77f945
QV2	contracts/swapV3periphery/lens/QuoterV2.sol	95fc092e8ec5972284bc149d0bd145a63e7bbdb4
TLE	contracts/swapV3periphery/lens/TickLens.sol	550febb5da62da41917d135142c2bb38657d52f0
NPM	contracts/swapV3periphery/NonfungiblePositionManager.sol	30f476030b4908d68175654ab4e32dc2ba30919c
TDE	contracts/swapV3periphery/TokenDescriptor.sol	b689818413720f4a06f2b6b338f7e1863558661e
IMU	contracts/swapV3periphery/interfaces/IMulticall.sol	be1f6b6df82eb4d03cb80d7cd9389d18009896e8
INPM	contracts/swapV3periphery/interfaces/INonfungiblePositionManager.sol	102e0b7ec30fca2c17b1a4c3cbddc22f50394cef
IQU	contracts/swapV3periphery/interfaces/IQuoter.sol	16c4d70ccffd3846eeb0a5868cab13f414ee7264
IPP	contracts/swapV3periphery/interfaces/IPeripheryPayments.sol	a7a35bcd0a9fb39238bf6da1b7387bc60ed7fade
IERC7P	contracts/swapV3periphery/interfaces/IERC721Permit.sol	f74a77ef10ca1abb1ce94d5ce850312123b2afec

IPI	contracts/swapV3periphery/interfaces/IPoolInitializer.sol	dbf06c69fe70eace743114ee77ca9a5ddf15b3c3
ISP	contracts/swapV3periphery/interfaces/ISelfPermit.sol	af0ab75a830bc0838f13ccaa581d3b7d015c4138
INTPD	contracts/swapV3periphery/interfaces/INonfungibleTokenPositionDescriptor.sol	f55edc89bcc171b7cebee510a527682d2a805726
ISR	contracts/swapV3periphery/interfaces/ISwapRouter.sol	a3189e961c8cd0a857894cfc2114b8f6f961c3f8
IPPWF	contracts/swapV3periphery/interfaces/IPeripheryPaymentsWithFee.sol	35e7cd079a1c8e74678c3eebc6ca16e186fc7ebe
IPIS	contracts/swapV3periphery/interfaces/IPeripheryImmutableState.sol	19d554a52a5ad9e52f99347410421f7acc23a074
ITL	contracts/swapV3periphery/interfaces/ITickLens.sol	e17e38c1e54f5f68b9f32f490443bed2755c5837
IERC2PA	contracts/swapV3periphery/interfaces/external/IERC20PermitAllowed.sol	057fc740dd796771ff66886730008494475135c1
IERC1	contracts/swapV3periphery/interfaces/external/IERC1271.sol	59fb7a09714ede65776a8d748553b215ca234d8a
IWETH9	contracts/swapV3periphery/interfaces/external/IWETH9.sol	903631286ba7625d9455c09810df336bbf428091
IQV2	contracts/swapV3periphery/interfaces/IQuoterV2.sol	a5f8c6c1c807475ef610b17c4534ba60e4b7c151

LMA	contracts/swapV3periphery/base/LiquidityManagement.sol	f4c8051a9bbffe7d50eb7cc53e8bf7a42ef3f477
MUL	contracts/swapV3periphery/base/Multicall.sol	e720a96e184f330b5d774493d021f2855814c4ce
PIN	contracts/swapV3periphery/base/PoolInitializer.sol	dfd574edf4c2b5da2dc540012d575c5a3bf0967c
BTI	contracts/swapV3periphery/base/BlockTimestamp.sol	c146bf4e466ece5399cf92a0c63410c8c0c78409
PPA	contracts/swapV3periphery/base/PeripheryPayments.sol	c48ec2f8366f306bfe9ab8af424dcdc98e3827dd
PPWF	contracts/swapV3periphery/base/PeripheryPaymentsWithFee.sol	22054a00490bdb70791224d488fbbbaf8cb054a1
ERC7P	contracts/swapV3periphery/base/ERC721Permit.sol	fa2eb7c19b6fcf9e40ca8ce332104ac4ccc90b33
PVA	contracts/swapV3periphery/base/PeripheryValidation.sol	0ac65ea78edb1043e0114ab838b453b2dbb0847b
SPE	contracts/swapV3periphery/base/SelfPermit.sol	aab588edca019028e92b88f95413a93180b4e382
PIS	contracts/swapV3periphery/base/PeripheryImmutableState.sol	1e25d844e88809d5e7473e165dc1a90ce8124fc3
HV2L	contracts/hyperindexV2/libraries/HyperindexV2Library.sol	6d97e1c826ba22be4948393603e42feb4c81c01b
MAT	contracts/hyperindexV2/libraries/Math.sol	7f5a442850545f05ec642203c248a6ae3a19456a

SMA	contracts/hyperindexV2/libraries/SafeMath.sol	9a37f7d9a06d2ad05fdc71a506433a4fedb9aebd
THE1	contracts/hyperindexV2/libraries/TransferHelper.sol	b2441f79a02b206ade7ff9e1b0f47be8f3b2e7f8
UQ1X1	contracts/hyperindexV2/libraries/UQ112x112.sol	a2aa89f19d5a1167fda2ade934d343a175bde994
HV2R0	contracts/hyperindexV2/HyperindexV2Router02.sol	ef53dbb4201974e0c85daddddb14fa39e4e1c378
IHV2C	contracts/hyperindexV2/interfaces/IHyperindexV2Callee.sol	31af00383b33f901576354b0e1e5f9e0a9966da8
IHV2P	contracts/hyperindexV2/interfaces/IHyperindexV2Pair.sol	31a72bca2239c4b21573dd80ff4bcf225171ef82
IHV2R0	contracts/hyperindexV2/interfaces/IHyperindexV2Router01.sol	e46892d90a3a8879913eb492065d6251b201db9a
IWETH1	contracts/hyperindexV2/interfaces/IWETH.sol	6f61b0bc2ca5baa63c38b3570aad51e356d5c25a
IHV2ERC2	contracts/hyperindexV2/interfaces/IHyperindexV2ERC20.sol	22534c8ec72129af8e6b43086934e3cd2a0e56ec
IHV2F	contracts/hyperindexV2/interfaces/IHyperindexV2Factory.sol	7053668824130be05935083e06875b5fddec372e
IERC2	contracts/hyperindexV2/interfaces/IERC20.sol	cfc1e58b14d7ed2f84c2b6fb684823b70214a4f9
IHV2R01	contracts/hyperindexV2/interfaces/IHyperindexV2Router02.sol	b1e424217cddbb03faf6c44a6482ea93e6fbd275

HV2ERC2	contracts/hyperindexV2/HyperindexV2ERC20.sol	d69f0df10700665af38c3ec53194eb0c8309862f
HV2F	contracts/hyperindexV2/HyperindexV2Factory.sol	ca920cfd13a01daad8740da7b7d861acd7f9df41
HV2P	contracts/hyperindexV2/HyperindexV2Pair.sol	69b113e44c907b3fe7abfe9522d03cbe3213375f
TIC	contracts/swapV3core/libraries/Tick.sol	4bb7da8e37ee6c6a5e6b54ba0041875662ce129a
SCA	contracts/swapV3core/libraries/SafeCast.sol	5b0caff4ab25e409eced395ee55c60c67555be5e
FP9	contracts/swapV3core/libraries/FixedPoint96.sol	f70ffcae9708509aa6e396a93f711d3e31b500fe
POS	contracts/swapV3core/libraries/Position.sol	ed36e74278d66b19e5b10d9d25b21f1cdceefd16
FMA	contracts/swapV3core/libraries/FullMath.sol	27a58d1447c4c8e58002408707e93343670a96a2
SMA1	contracts/swapV3core/libraries/SwapMath.sol	bd92a5e3ce969ee76c0f3131b6253322372e1221
ORA	contracts/swapV3core/libraries/Oracle.sol	86f564d90271e9eeb9955043cf25dea9a8038297
FP1	contracts/swapV3core/libraries/FixedPoint128.sol	6ef3cb38c19081095f2842df6c80c85504001be5
THE2	contracts/swapV3core/libraries/TransferHelper.sol	83db09a8fada750beaddfd248289b43235447d01

TBI	contracts/swapV3core/libraries/TickBitmap.sol	47f5ce918263012ada294edaf99b902639cf6d66
TMA	contracts/swapV3core/libraries/TickMath.sol	4df139f4313a35789b1a391296862cbb28ba6e7b
UMA	contracts/swapV3core/libraries/UnsafeMath.sol	cc232c793953dc31a9fd52da1a98c04e528c280e
BMA	contracts/swapV3core/libraries/BitMath.sol	c4525c6bda29b6fc974daa704488c8ab29e4e3d7
SPM	contracts/swapV3core/libraries/SqrtPriceMath.sol	dcf3fbe26b6e2fa3379ed66fb3022f56fd04107a
NDC	contracts/swapV3core/NoDelegateCall.sol	51831fabcc45cf427cd70d9ea9202503d49816a2
HV3P	contracts/swapV3core/HyperindexV3Pool.sol	3c65d9ecc84809fbac145c130e4a4fa66a3c95e2
ISV3F	contracts/swapV3core/interfaces/ISwapV3Factory.sol	0b103bddbd93a31ce87d6c092267120da83a766e
IERC2M	contracts/swapV3core/interfaces/IERC20Minimal.sol	6f563f63c5f24b26d7278d296b0c93b9dfa2fc8d
ISV3PD	contracts/swapV3core/interfaces/ISwapV3PoolDeployer.sol	8c7fe14acd6962228fe4c01b25b8d0058f52bdaf
ISV3SC	contracts/swapV3core/interfaces/callback/ISwapV3SwapCallback.sol	ee7d595d3ee5fab89e87a785b98517e1d117756d
ISV3MC	contracts/swapV3core/interfaces/callback/ISwapV3MintCallback.sol	2de63682622b6904143f76edc82239e4320f565a

ISV3FC	contracts/swapV3core/interfaces/callback/ISwapV3FlashCallback.sol	e2b3e7ff0082dd4a3a3f32f5c4a1731997d234b8
ISV3PI	contracts/swapV3core/interfaces/pool/ISwapV3PoolImmutables.sol	23afbe84fc19700fd6dfd9bf28f216344599f76b
ISV3PDS	contracts/swapV3core/interfaces/pool/ISwapV3PoolDerivedState.sol	97a5287b75618ad0c430d08d8617902f0d3a6e29
ISV3PA	contracts/swapV3core/interfaces/pool/ISwapV3PoolActions.sol	0c4ef74d811a2e5c438c9d176816ebc75de398db
ISV3PE	contracts/swapV3core/interfaces/pool/ISwapV3PoolEvents.sol	9864b9339afcf982ba21a83d2a1efd4a75b3e950
ISV3PE1	contracts/swapV3core/interfaces/pool/ISwapV3PoolErrors.sol	b3fcea556ca1aa0a0c7839e169e771f072cd1511
ISV3PS	contracts/swapV3core/interfaces/pool/ISwapV3PoolState.sol	dfa6127aba2920c274c33646f62fa941c1ef480d
ISV3POA	contracts/swapV3core/interfaces/pool/ISwapV3PoolOwnerActions.sol	301a1e5d8dd7ff346412a117321791ba009f01e2
ISV3P	contracts/swapV3core/interfaces/ISwapV3Pool.sol	8fa12ab726e2243eb833b10943b8577f15114379
HV3F	contracts/swapV3core/HyperindexV3Factory.sol	1b6b991b060658b0cf55fb81deed4e5eaf18abd1
SV3PD	contracts/swapV3core/SwapV3PoolDeployer.sol	a1a32539ceb732a6e5048188c539cfc4b2f00183

1.3 Issue Statistic

Item	Count	Fixed	Acknowledged
Total	3	0	3
Informational	2	0	2
Minor	0	0	0
Medium	0	0	0
Major	0	0	0
Critical	0	0	0

1.4 ScaleBit Audit Breakdown

ScaleBit aims to assess repositories for security-related issues, code quality, and compliance with specifications and best practices. Possible issues our team looked for included (but are not limited to):

- Transaction-ordering dependence
- Timestamp dependence
- Integer overflow/underflow
- Number of rounding errors
- Unchecked External Call
- Unchecked CALL Return Values
- Functionality Checks
- Reentrancy
- Denial of service / logical oversights
- Access control
- Centralization of power
- Business logic issues
- Gas usage
- Fallback function usage
- tx.origin authentication
- Replay attacks
- Coding style issues

1.5 Methodology

The security team adopted the **"Testing and Automated Analysis"**, **"Code Review"** and **"Formal Verification"** strategy to perform a complete security test on the code in a way that is closest to the real attack. The main entrance and scope of security testing are stated in the conventions in the "Audit Objective", which can expand to contexts beyond the scope according to the actual testing needs. The main types of this security audit include:

(1) Testing and Automated Analysis

Items to check: state consistency / failure rollback / unit testing / value overflows / parameter verification / unhandled errors / boundary checking / coding specifications.

(2) Code Review

The code scope is illustrated in section 1.2.

(3) Audit Process

- Carry out relevant security tests on the testnet or the mainnet;
- If there are any questions during the audit process, communicate with the code owner in time. The code owners should actively cooperate (this might include providing the latest stable source code, relevant deployment scripts or methods, transaction signature scripts, exchange docking schemes, etc.);
- The necessary information during the audit process will be well documented for both the audit team and the code owner in a timely manner.

2 Summary

This report has been commissioned by PotatoSwap to identify any potential issues and vulnerabilities in the source code of the PotatoSwap smart contract, as well as any contract dependencies that were not part of an officially recognized library. In this audit, we have utilized various techniques, including manual code review and static analysis, to identify potential vulnerabilities and security issues.

During the audit, we identified 3 issues of varying severity, listed below.

ID	Title	Severity	Status
HV2-1	Insecure Ownership Transfer Pattern in `setFeeToSetter` Function	Informational	Acknowledged
HV3-1	Potential Risk Adding New Fee Tier	Informational	Acknowledged
MU2-1	Inconsistent Implementation of `getCurrentBlockDifficulty()` with On-Chain Version	Discussion	Acknowledged

3 Participant Process

Here are the relevant actors with their respective abilities within the PotatoSwap Smart Contract :

Factory Owner

- setOwner: Changes the owner of the factory contract.
- enableFeeAmount: Enables a new fee tier and corresponding tick spacing.
- setFeeProtocol: Sets the protocol fee ratio (called through the pool contract).
- collectProtocol: Collects protocol fees (called through the pool contract).

Liquidity Provider

- NonfungiblePositionManager.mint: Creates a new liquidity position and mints an NFT.
- NonfungiblePositionManager.increaseLiquidity: Increases the liquidity of an existing position.
- NonfungiblePositionManager.decreaseLiquidity: Decreases the liquidity of an existing position.
- NonfungiblePositionManager.collect: Collects the accumulated trading fees.
- NonfungiblePositionManager.burn: Burns the NFT position (must remove all liquidity first).

Trader

- HyperindexV3Router.exactInputSingle: Executes a single-pool swap with an exact input amount.
- HyperindexV3Router.exactInput: Executes a multi-pool path swap with an exact input amount.
- HyperindexV3Router.exactOutputSingle: Executes a single-pool swap with an exact output amount.
- HyperindexV3Router.exactOutput: Executes a multi-pool path swap with an exact output amount.

NFT Holder

- NonfungiblePositionManager.approve: Authorizes another address to operate the NFT.
- NonfungiblePositionManager.transferFrom: Transfers ownership of the NFT.
- NonfungiblePositionManager.safeTransferFrom: Safely transfers ownership of the NFT.
- NonfungiblePositionManager.permit: Grants NFT operation permission via signature authorization.

Flash Borrower

- HyperindexV3Pool.flash: Executes a flash loan.

Any User

- Various read/query functions, such as factory(), getPool(), positions(), observations(), etc.
- HyperindexV3Pool.initialize: Initializes a newly created pool (can only be called once).

4 Findings

HV2-1 Insecure Ownership Transfer Pattern in `setFeeToSetter` Function

Severity: Informational

Status: Acknowledged

Code Location:

contracts/hyperindexV2/HyperindexV2Factory.sol#49-52

Descriptions:

The `HyperindexV2Factory` contract uses a direct setter pattern for changing the `feeToSetter` role, which can lead to accidental loss of control. In the current implementation, the `setFeeToSetter` function allows the current `feeToSetter` to directly set a new address without any confirmation from the new owner.

```
function setFeeToSetter(address _feeToSetter) external override {
    require(msg.sender == feeToSetter, 'HyperindexV2: FORBIDDEN');
    feeToSetter = _feeToSetter;
}
```

Suggestion:

Implement a two-step transfer pattern (transfer-accept) similar to how ownership is handled in more secure implementations. This pattern involves:

The current `feeToSetter` proposes a new `feeToSetter` (transfer) The proposed new `feeToSetter` must explicitly accept the role (accept) Example Implementation

```
// ... existing code ...
    address public override feeTo;
    address public override feeToSetter;
    address public pendingFeeToSetter;

// ... existing code ...
```

```
    function setFeeToSetter(address _feeToSetter) external override {
        require(msg.sender == feeToSetter, 'HyperindexV2: FORBIDDEN');
        pendingFeeToSetter = _feeToSetter;
    }

    function acceptFeeToSetter() external {
        require(msg.sender == pendingFeeToSetter, 'HyperindexV2: FORBIDDEN');
        feeToSetter = pendingFeeToSetter;
        pendingFeeToSetter = address(0);
    }

// ... existing code ...
```

This approach ensures that:

The role cannot be accidentally transferred to an incorrect address. The new `feeToSetter` must explicitly accept the role. There is no risk of permanently losing the role due to typos or transferring to non-handling contracts.

HV3-1 Potential Risk Adding New Fee Tier

Severity: Informational

Status: Acknowledged

Code Location:

contracts/swapV3core/HyperindexV3Factory.sol

Descriptions:

Adding a new 0.01% fee tier (Tick Spacing=2) to the Uniswap V3 factory contract introduces triple risks in protocol compatibility, economic rationality, and technical security. Specifically: peripheral contracts (e.g., routers and managers) will fail to recognize the new fee tier if not updated simultaneously, leading to liquidity operation failures; excessively low fees may inadequately compensate market makers for risks in non-stablecoin pairs, while fragmented liquidity could increase slippage; additionally, the reduced tick spacing significantly increases gas costs, and unauthorized code modifications may introduce security vulnerabilities without rigorous auditing.

Suggestion:

Before deployment, comprehensive end-to-end testing (covering pool creation, liquidity operations, and trading scenarios) must be completed, with all peripheral contracts and frontend interfaces updated to support the new fee tier. Furthermore, modified contracts should undergo strict auditing by professional firms. Continuous monitoring of liquidity depth and strategic incentivization are essential to guide capital allocation, ensuring the new fee tier achieves an optimal balance between low slippage and sustainable market maker returns.

MU2-1 Inconsistent Implementation of `getCurrentBlockDifficulty()` with On-Chain Version

Severity: Discussion

Status: Acknowledged

Code Location:

contracts/help/Multicall2.sol#36-38

Descriptions:

In this contract implementation, the function `getCurrentBlockDifficulty()` returns `block.prevrandao`. However, the on-chain version of Multicall2 (deployed on Ethereum mainnet) uses `block.difficulty` instead.

Suggestion:

Determine which one to choose here.

Resolution:

This issue has been fixed. The client has adopted our suggestions.

Appendix 1

Issue Level

- **Informational** issues are often recommendations to improve the style of the code or to optimize code that does not affect the overall functionality.
- **Minor** issues are general suggestions relevant to best practices and readability. They don't post any direct risk. Developers are encouraged to fix them.
- **Medium** issues are non-exploitable problems and not security vulnerabilities. They should be fixed unless there is a specific reason not to.
- **Major** issues are security vulnerabilities. They put a portion of users' sensitive information at risk, and often are not directly exploitable. All major issues should be fixed.
- **Critical** issues are directly exploitable security vulnerabilities. They put users' sensitive information at risk. All critical issues should be fixed.

Issue Status

- **Fixed:** The issue has been resolved.
- **Partially Fixed:** The issue has been partially resolved.
- **Acknowledged:** The issue has been acknowledged by the code owner, and the code owner confirms it's as designed, and decides to keep it.

Appendix 2

Disclaimer

This report is based on the scope of materials and documents provided, with a limited review at the time provided. Results may not be complete and do not include all vulnerabilities. The review and this report are provided on an as-is, where-is, and as-available basis. You agree that your access and/or use, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your own risk. A report does not imply an endorsement of any particular project or team, nor does it guarantee its security. These reports should not be relied upon in any way by any third party, including for the purpose of making any decision to buy or sell products, services, or any other assets. TO THE FULLEST EXTENT PERMITTED BY LAW, WE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, IN CONNECTION WITH THIS REPORT, ITS CONTENT, RELATED SERVICES AND PRODUCTS, AND YOUR USE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NOT INFRINGEMENT.