

Zxo on Arbitrum Deployment Statement

VERIFIED SOURCE · EXACT MATCH

DATE	BASED ON	COMPILER	BUILD SETTINGS
2 October 2026	Security Audit Revision 3.0	solc 0.8.36+commit.8a079791	Optimizer 200 runs · Shanghai · via-IR off
VERIFICATION	REVIEWER		
Sourcify, creation and runtime bytecode	Claude (Anthropic), AI-assisted		

01 Statement

AUDIT SCORE

8.7/10

8.5 → 8.7 in Revision 3.0

Score of the audited code as deployed.
It is not a rating of the tokens, the team or any investment.

The Zxo contracts deployed on Arbitrum One at the five addresses in section 02 are built from the source code reviewed in Zxo Security Audit Revision 2.5. Sourcify reports an exact match of creation and runtime bytecode for each contract.

The deployed source differs from the reviewed files in three places only: the BUSL-1.1 license header added to each file, the ChainLib import (now `ArbitrumLib.sol`), and the Arbitrum addresses and HOLDER settings inside `ArbitrumLib.sol`.

No audit finding is open in the deployed code, and no path was found that lets anyone take user principal. Unstake has no price gate and no owner gate, and the core LP has no withdrawal function.

Three operational items remain open (section 07): the owner is a single wallet, no static-analysis or third-party audit report was provided to this review, and the PancakeSwap and USDT paths have not yet run on their own chains.

5Contracts with
exact-match
source**6**Pools launched
in the enforced
order**58**Zxo transactions
reviewed**0**Differences from
audited logic

02

Deployed contracts

CONTRACT	ADDRESS (LINKS TO SOURCIFY)	BLOCK	VERIFIED (UTC)	BYTECODE
ZxoTreasury Fee and referral receiver	0x0341A7a6CC031662526492e10796E551125AE2bc	509546642	29 Sep 2026 11:17	EXACT MATCH
ZxoOracle 15-minute average price reader	0x98c9695bb012A42FDcB734324A78E63eE6c23D44	509547888	29 Sep 2026 11:19	EXACT MATCH
ZxoV3Deployer Seeds other DEXes, then finalizes	0xb6CB455CB67d59DD269C865428acece75eA08594	509561805	29 Sep 2026 11:21	EXACT MATCH
Zxo Farm, staking, rewards, core LP	0x5356B3004D0394dEA51161d175265EC8F92c96b7	509563331	29 Sep 2026 11:22	EXACT MATCH
ZxoToken (pool 0) Reward token of pool 0	0x304441b6BDADA52d41f4AA0A0b13dC4185B3979B	509698291	2 Oct 2026 17:07	EXACT MATCH

All five were deployed by 0xFf9623ABe5D081c3c4E586330ab714FcB718fa53, an externally owned account that is also the current owner of Zxo. The order on-chain was ZxoTreasury, ZxoOracle, ZxoV3Deployer, then Zxo (28 September 2026, 01:49 UTC), followed by `ZxoV3Deployer.setZxoContract(Zxo)` at block 509696927. That is the order the audit requires. Pools 1 to 5 use further ZxoToken instances created by `Zxo.createPool` from the same token code; only the pool 0 token was submitted for verification.

Deployed source against audited source

HASH MATCH keccak256 in the Sourcify metadata equals the hash of the reviewed file

LINE MATCH verified source compared line by line with the reviewed file

FILE	METHOD	KECCAK256 OF DEPLOYED FILE	RESULT
contracts/ZxoTreasury.sol	HASH MATCH	0x0079a87d4d42342400d69e77edfdef134ec908260c99c68f81747a2f0a213056	Identical to the reviewed file
contracts/ZxoOracle.sol	HASH MATCH	0x8afb8ab8cd5759e84829fb1de90d6b019e8c05ccb3029a9443d2906d5a4056c69	Identical to the reviewed file
contracts/tokens/ZxoToken.sol	HASH MATCH	0x672fc398591453b8e92a5781c8ccdefeb42b4efa718056bc802927f95ee470c1	Identical to the reviewed file
contracts/ZxoV3Deployer.sol	HASH MATCH	0x18724bad994f0598fa5e063cf9604db3c932e3a5f059c0af16cc2de659492b7d	Identical; import switched to ArbitrumLib
contracts/interfaces/IZxo.sol	HASH MATCH	0xe5829a468f8af5b9c339293d8360b5cdc533d65d24fb79efc147569868508a0f	Identical to the reviewed file
contracts/interfaces/Iv3PosManager.sol	HASH MATCH	0x982265f349185d696df1f26f1954c203c0ac4bc7498ae34b2001edd61dc1a917	Identical to the reviewed file
contracts/libraries/ArbitrumLib.sol as compiled into ZxoV3Deployer	HASH MATCH	0xba0c9c8350ceab6d746617a4c95cae995f726174ffd17f49ba321a83a7813a5e	Treasury and oracle set, deployer still zero, HOLDER settings as in section 04

FILE	METHOD	KECCAK256 OF DEPLOYED FILE	RESULT
contracts/Zxo.sol	LINE MATCH	0x609ab4d98b06c234f577e1a40d19587dc7d16ed5074b1c44cd9ccac05924570f expected value, see note	907 lines compared, no code difference; import switched to ArbitrumLib
contracts/libraries/ZxoLib.sol	LINE MATCH	0x6350d7aa68bfe68ab1c2ff95c70db68c621a96b132f376374900b6d4d61ef30d expected value, see note	428 lines compared, no difference
contracts/libraries/ArbitrumLib.sol as compiled into Zxo	LINE MATCH	0x29e77369880de3bf9d809129c826367eb38f2ccf719d63974f2721036b18fb75 expected value, see note	All three Zxo addresses set; otherwise as above
contracts/interfaces/IZxoToken.sol	LINE MATCH	0x329833fbb1620d4412527a8e9f462f56694ee01169787d81532a35db6d4801c8 expected value, see note	No difference
contracts/interfaces/IZxoOracle.sol	LINE MATCH	0x844e984436f41bcec3ad7b0ba0e06b4126dbd149a8de1b79f9c232a9207f1cec expected value, see note	No difference
contracts/interfaces/IZxoV3Deployer.sol	LINE MATCH	0x32a366db0f56d1ae5489bc7121d87d0d2809d3d39ef5f9949ce56f0c912ee463 expected value, see note	No difference

Hash match. The compiler metadata embedded in each deployment lists a keccak256 hash for every source file. For the files marked Hash match, that hash equals the hash of the reviewed file with the published license header (Change Date 2029-09-28) placed in front. This is an exact, byte-level result.

Line match. The Zxo metadata was too large for this review's reader to reach the hashes of `Zxo.sol` and `ZxoLib.sol`. Those files were instead read from the Sourcify-verified input and compared with the reviewed files line by line, ignoring whitespace. No code line differs. The hash shown for these files is the value the reviewed file produces; anyone can confirm it against `metadata.json` on Sourcify to turn the line match into a hash match.

Interface. The verified ABI of Zxo has 24 functions, 3 events and 4 errors, and matches the reviewed interface, including `unstake(uint64,uint8,bool,address)` and the single `ZxoEvent`. The OpenZeppelin files carry identical hashes in all five deployments; they were not compared with an upstream release.

04

Configuration compiled into Zxo

CONSTANT (ARBITRUMLIB)	VALUE	NOTE
CHAIN_ID	42161	Arbitrum One
POS_MANAGER	0xC36442b4a4522E871399CD717aBDD847Ab11FE88	Uniswap V3 position manager
FACTORY	0x1F98431c8aD98523631AE4a59f267346ea31F984	Uniswap V3 factory
ZX0_TREASURY	0x0341A7a6CC031662526492e10796E551125AE2bc	Matches section 02
ZX0_V3_DEPLOYER	0xb6CB455CB67d59DD269C865428acece75eA08594	Matches section 02
ZX0_ORACLE	0x98c9695bb012A42FDcB734324A78E63eE6c23D44	Matches section 02
HOLDER_TOKEN	0x6F296078e80907E23014a619C83d98912E35E66A	Changed since Revision 2.5, which reviewed 0x286e...F164
HOLDER_THRESHOLD	3,690,000,000 tokens	Changed since Revision 2.5, which reviewed 369,000

What a successful Zxo deployment proves

The Zxo constructor reverts unless every check passes, so the deployment at block 509563331 shows that on that block: the chain id was 42161; all six configured addresses held code; the position manager used the configured factory; the 0.01% fee tier had tick spacing 1; the HOLDER token reported 18 decimals and the threshold was at least one token; the treasury and deployer returned their identity markers; the deployer was not yet linked to another Zxo; and the oracle reported its 15-minute window. It also shows the contract compiled without stack errors and fits the 24,576-byte size limit.

The HOLDER token and threshold are not security settings, but they decide who gets the extra 25% bonus. The team should confirm that the new token and the 3.69 billion threshold are the intended ones. The test stakes reviewed below received the standard bonus without the HOLDER increase.

On-chain activity reviewed

Events and transactions for Zxo and ZxoV3Deployer were read through the Arbitrum Blockscout API, from deployment to 1 October 2026, 11:23 UTC: 58 Zxo transactions with 95 decoded `ZxoEvent` logs, and the first 14 ZxoV3Deployer transactions. Every transaction listed succeeded.

Functions exercised on mainnet

OPERATION	SEEN	WHAT IT CONFIRMS
Pool launch: createPool, core LP, seed, finalize, core stake	6 pools	Each pool followed the enforced order. Every reward token was finalized before its core stake was minted.
Seeding on another DEX	6	All through position manager <code>0x46A15B0b27311cedF172AB29E4f4766fbE7F4364</code> , the PancakeSwap V3 position manager address. Confirm it against PancakeSwap's published Arbitrum addresses.
Stake	11	The health gate passed, so the spot price and the 15-minute average were both read and in range on live Uniswap V3 pools.
Harvest (claim and collect)	8 + 8	Rewards were minted on claim and trading fees were forwarded.
Unstake	7	Each position was held 24.6 to 37.4 hours, past the 1-day lock-up. Forfeited pending rewards were logged, not minted.
Config changes and rate re-targeting	10 + 20	Target APR and referral rate changes logged the stored value; the emission rate re-targeted after each.
ZxoV3Deployer set-up	1	The validator was set to the zero address at deployment, which disables the testnet-only token path. <code>setZxoContract</code> was called once.

Numbers recomputed from the audited formulas

CHECK (POOL 0)	FORMULA RESULT	ON-CHAIN VALUE	RESULT
First reward claim: $\text{rate} \times \text{seconds} \times \text{weight} \div \text{total weight}$, over 2,351 seconds	3,583,870,758 wei	3,583,870,758 wei	EXACT
Staked value counted for emissions with only the core stake present	0.187688 tokens	0.187688 tokens	MATCH

CHECK (POOL 0)	FORMULA RESULT	ON-CHAIN VALUE	RESULT
Stake bonus at the launch price	10% to 12% expected	10.99% on all 11 stakes	IN RANGE
Core LP size from its liquidity over ticks 11 to 300	361,486.49 tokens	liquidity 2.52×10^{25}	DERIVED

The first row reproduces a live claim to the last wei from the reward formula in the audit. The second shows that a single-sided stake counts toward the value used for emissions, which was finding M-01. The staked value on-chain comes from the emission rate set by the first re-target; the formula result comes from the core-stake liquidity and its tick range.

All activity so far is the team's own

Every stake, claim and unstake in the period came from the deployer address. This is a live mainnet test by the team with small amounts, not evidence of use by the public. It does show the contracts behave as audited on Arbitrum.

06

Pools and owner settings observed

POOL	REWARD TOKEN	CREATED (UTC)	MAX TOKENS PER 30 DAYS	TARGET APR AT 1 OCT	BURNED AT FINALIZE
0	0x304441b6BDADA52d41f4AA0A0b13dC4185B3979B	28 Sep 11:57	37.54	640%	3.138138 tokens
1	0x113CA9e137ee2FD5bB89C979255010BBb3997D4f	28 Sep 16:38	12,010,569,300.98	100%	0
2	0x387BAfD20705c33Ff04A392C125BDeB39DfC6f98	28 Sep 18:37	485,201.36	75%	0
3	0x1BABe35Eb60B5bdd4E4299EBdc6d511E29F0E906	28 Sep 20:44	100,000	32%	453,914 wei
4	0x3Ec1c1ED083156c2B33Bf49550eCc9390e1b50eB	28 Sep 23:51	4,947.23	220%	0
5	0x0cc4C9B093223E95d93FF10E6C6CdEae14579258	29 Sep 13:48	1.04	240%	92,483 wei

- Every pool was created with the highest permitted cap, `maxNormalAPR` = 320%, which doubles to 640% in the first year. Pool 0's target is set to that 640% ceiling.

- Actual emissions are also limited by each pool's 30-day cap, which is fixed at creation and cannot be raised. Pool 0 can mint at most 37.54 tokens per 30 days.
- The referral rate was raised to the 30% maximum on pools 0 and 4. The referral share comes out of each claim, so stakers on those pools keep 70% of their rewards unless they are their own referrer.
- Pool 0's core stake is about 0.19 tokens beside a 361,486-token core LP. At that size the core-stake dilution described in finding M-03 is negligible once users stake more than a few tokens.
- At least three `createPool` calls used `withoutPool = true`. These deploy a token with no pool and no supply and have no effect on the pools.

These settings are within the limits the contract enforces and are the owner's to choose. They are listed so that users can see them; they are not findings.

07

Open items

OPEN not done or not evidenced **TEAM-REPORTED** stated by the team, not independently checked

ITEM	STATUS	DETAIL
Owner is a single wallet	OPEN	Zxo and ZxoV3Deployer are owned by the deployer account. No ownership transfer appears in the logs. The team plans a multisig after all mainnet deployments. Until then one key can create pools, change target APR and referral rates, and collect core LP fees. It cannot touch staked positions, block unstaking or withdraw core liquidity.
Static analysis (Slither or Aderyn)	OPEN	The team's note reports successful testing but does not say a static analyzer was run. No report was provided.
Independent human audit	TEAM-REPORTED	The team states that all audits passed. No third-party report was provided to this review, so none is relied on here.
PancakeSwap pool as the main pool; USDT pair	OPEN	On Arbitrum the main pools are Uniswap V3. Reading the price from a PancakeSwap pool (BNB Chain) and staking a USDT pair (Ethereum) have not run on-chain yet. Test both on a fork before those deployments.
Revert tests	TEAM-REPORTED	Pre-initialized pool, mismatched decimals, and unstake to a contract that cannot receive NFTs. The team confirms each reverts as designed. A contract staker must be able to pass the <code>to</code> argument to recover.
Price history per pool	TEAM-REPORTED	Stakes and harvests succeeded, so the 15-minute average was available. The stored history size of each pool was not read. Run <code>set-observations.py</code> for all six pools if not already done.

ITEM	STATUS	DETAIL
HOLDER settings	OPEN	Confirm the new HOLDER token and 3.69 billion threshold are intended (section 04).

08

Limits of this statement

- It covers the five addresses in section 02 on Arbitrum One only. Other chains need their own statement.
- Source and bytecode status come from Sourcify. Events and transactions come from the Arbitrum Blockscout API, up to 1 October 2026, 11:23 UTC. Later activity is not covered.
- No contract state was read directly from a node. The current owner, the deployer link, pool history sizes and treasury settings are inferred from events, not read from storage.
- `Zxo.sol` and `ZxoLib.sol` were matched line by line, not by hash (section 03).
- `ZxoTreasury` ownership and destination, and any `ZxoVerify` deployment, were not reviewed on-chain.
- No fuzzing, static analysis or fork testing was run by this review. This AI-assisted statement does not replace an independent human audit.

09

Verify it yourself

Source and bytecode

- Open each address in section 02 on Sourcify and check that it shows Exact Match.
- Download `metadata.json` for `Zxo` and compare the `keccak256` of `contracts/Zxo.sol` and `contracts/libraries/ZxoLib.sol` with section 03.
- Or compile the published standard JSON input with solc 0.8.36 and compare the runtime bytecode with the chain.

Before you stake in a pool

- `ZxoV3Deployer.isFinalized(rewardToken)` returns true.
- The deployer's balance of the reward token is 0.
- Each `PoolStarted` event names a position manager you recognize.
- `Zxo.owner()` shows who controls pool settings; check whether it is a multisig yet.
- `Zxo.poolInfo(id)` shows the target APR and referral rate that apply to you.

This statement is an AI-assisted review of publicly verified source code and on-chain records for the addresses listed. It is not a guarantee that the code is free of defects, and it is not financial or legal advice. Behavior depends on the external DEX contracts, chain conditions and the owner's settings. An independent professional audit is recommended for contracts that hold user funds.