CryptoAudit.PRO
INVESTIGATIONS & AUDITS

WindfallLotto

Smart Contract Security Assessment

ECOSYSTEM	Polygon (POL) Mainnet
LANGUAGE	Solidity v0.8.31+commit.fd3a2265
METHODS	Bytecode Analysis · ABI Review · External Tool Evaluation
ASSESSED BY	CryptoAudit.PRO — Investigations & Audits
DATE	May 5, 2026
REPORT VERSION	v2.2 — Final (on-chain verification complete)

CONTRACT ADDRESS

`0x9650D206c6e0093FBc1D623b2A1e03984D24d3f1`

0	0	4	5	6	0
Critical	Major	Medium	Minor	Info	Central.

15 TOTAL FINDINGS | 0 RESOLVED | 15 OPEN | 4 RECLASSIFIED | WFL-02 VERIFIED ON-CHAIN

Table of Contents

★ = New finding validated from external tool review

Amendment Change Log — v2.1 / On-Chain Verification — v2.2

v2.1 Amendment Notice: Four findings reclassified following formal client response (WFL-01 Major→Medium, WFL-02 Centralization→Informational, WFL-03 Centralization→Informational, WFL-07 Medium→Minor). **v2.2 Verification Notice:** WFL-02 access model independently verified on-chain (May 8, 2026). Two distinct non-owner addresses successfully executed draw lifecycle functions for Draw #5, confirming the permissionless design. A new positive observation has been added for the MetaMask Delegation Toolkit integration.

Finding	v2.0	v2.1 / v2.2	Decision	Evidence / Rationale
WFL-01	Major Blockhash Randomness	**Medium Last-Resort Blockhash Fallback**	**Partial Accept**	Tertiary fallback requires two prior failures. Liveness tradeoff acknowledged. Major overstates probability; Medium retained for consequence severity.
WFL-02	Centralization Centralized Draw Lifecycle	**Informational Operational Reliance on Host Execution**	**Accepted Verified On-Chain**	Permissionless access confirmed. 0x46c5642c (via MetaMask delegation) called closeDrawAndRequestRandom; 0x87A76864 called processTickets directly on WindfallLotto for Draw #5. Both succeeded. Functions carry no onlyOwner guard.
WFL-03	Centralization Shared Treasury/Supra Key	**Informational Operational Key Separation**	**Accepted**	Shared key does not enable jackpot theft, ticket manipulation, or user exclusion. Risk is key management concentration only. Multisig migration planned by team.
WFL-07	Medium VRF Revert Bypasses Fallback	**Minor VRF Funding Dependency Before Fallback**	**Partial Accept**	VRF subscription fundable externally by anyone; not a permanent lockout. Architectural bypass observation retained as design note at lower severity.

1. Executive Summary

This security assessment was prepared by **CryptoAudit.PRO** for the WindfallLotto protocol — a decentralized on-chain lottery deployed on Polygon Mainnet. The assessment was conducted using ABI analysis, bytecode review, constructor argument decoding, live transaction history inspection, and critical evaluation of third-party automated scanning outputs from SolidityScan and AuditBase.

A total of **15 findings** were identified. **No Critical vulnerabilities were found.** This report is version **v2.1**, amended following a formal client The final severity profile is: 0 Critical, 0 Major, 4 Medium, 5 Minor, 6 Informational, 0 Centralization.

Category	Count	Highest Severity	Status
Major	0	— (reclassified to Medium in v2.1)	Closed
Medium	4	Last-resort blockhash fallback; no emergency recovery	Open
Minor	5	Gas limit; whale dominance; tier validation; VRF liveness	Open
Informational	6	Confirmations; pause; Supra trust; NatSpec; operational	Open
Centralization	0	— (reclassified to Informational in v2.1)	Closed

2. Project Information

Project Name	WindfallLotto
Contract Name	WindfallLotto
Network	Polygon Mainnet (Chain ID: 137)
Contract Address	0x9650D206c6e0093FBc1D623b2A1e03984D24d3f1
Compiler	v0.8.31+commit.fd3a2265 | Optimization: 200 runs
Token (Payment)	DAI (PoS) 0x8f3Cf7ad23Cd3CaDbD9735AFf958023239c6A063
Deployer / Owner	0x0d18342127412658AEf47bA799C27259c4C17268 (windfall-lotto.eth)
Contract Age	~30 days at time of assessment
Total Transactions	57 confirmed transactions

■ Website	https://windfall-lotto.com/
✘ Twitter / X	https://x.com/WindfallPolygon
■ YouTube	https://www.youtube.com/@windfall-lotto-eth
Telegram	https://t.me/windfalllottogroup
Discord	https://discord.gg/XnAupfE6BX
OpenSea	https://opensea.io/collection/windfall-lotto-ticket

3. Audit Scope & Codebase

File	Description
WindfallLotto.sol	Main lottery contract — draw lifecycle, randomness, prizes, fee distribution
WindfallTicket.sol	ERC-721 lottery ticket NFT with tier assignment (onlyLotto)
WindfallDrawNFT.sol	ERC-721 draw record NFT minted per completed draw
WindfallSVG.sol	On-chain SVG generation for ticket and draw NFT artwork
@openzeppelin/...	IERC20, SafeERC20, ERC721, ERC721Enumerable, ReentrancyGuard, Base64
@chainlink/...	VRFConsumerBaseV2Plus, ConfirmedOwner, IVRFCoordinatorV2Plus, VRFV2PlusClient

Contract	Address
WindfallTicket NFT	`0x8A1E8B8c54338bAa7B239dB845316A37BCb07C41`
WindfallDrawNFT	`0x120C9ce64cfd6A2B173A6B44dc6aCFA5fEB556c1`
FeeShare	`0x8d1e76657F469932Dd04d0Bad2f0FCE0bbDb22a5`
VRF Coordinator	`0xec0Ed46f36576541C75739E915ADbCb3DE24bD77`
Supra Router	`0x76606cD35d3De51d2c2e44D6eb7AF593D8dfD983`

4. Approach & Methods

This assessment was conducted using a comprehensive multi-method approach given that the Solidity source was submitted in Standard JSON-Input format.

ABI Analysis	Full review of 65+ functions, events, and custom error types to reconstruct contract logic, access control patterns, and fund flows.
Bytecode Review	Deployed bytecode analysis to extract hardcoded constants (TICKET_PRICE, BPS values, gas limits, timeouts) and verify constructor-set immutables.
Constructor Argument Decoding	All 14 constructor arguments decoded and cross-referenced to identify address relationships and configuration risks.
Transaction History Review	All 57 on-chain transactions analyzed to verify operational patterns, identify who calls which functions, and confirm draw lifecycle behaviour.
External Tool Evaluation	SolidityScan (22 checks) and AuditBase AI report (6 findings) reviewed, with each finding independently validated or dismissed against actual bytecode.
Randomness Architecture Analysis	Full three-tier randomness chain (Chainlink VRF → Supra → Blockhash) modelled for failure modes, timeout conditions, and manipulation vectors.

5. Vulnerability Summary

A total of 15 findings were identified. No critical vulnerabilities were found. Four findings were reclassified in v2.1; WFL-02 was verified on-chain in v2.2 (marked ▲). The two highest-priority items remain WFL-01 (blockhash randomness bias) and WFL-07 (VRF funding dependency bypassing fallback chain).

ID	Title	Severity	Category	Status	Source
WFL-01	Last-Resort Blockhash Fallback — Conditional Trust Assumption ▲	Medium	Volatile Code	Open	Primary
WFL-02	Operational Reliance on Host Execution — Verified Permissionless ▲	Informational	Operational Risk	Verified	Primary
WFL-03	Operational Key Separation Recommendation ▲	Informational	Centralization	Open	Primary
WFL-04	No Emergency Fund Recovery Mechanism	Medium	Volatile Code	Open	Primary
WFL-05	Ticket Price Immutable — No Market Peg	Medium	Volatile Code	Open	Primary
WFL-06	BPS Values Lack Constructor Sum Invariant	Medium	Coding Issue	Open	Primary
WFL-07	VRF Funding Dependency Before Fallback Activation ▲	Minor	Volatile Code	Open	AuditBase ★
WFL-08	VRF Callback Gas Limit Very High	Minor	Gas Optimization	Open	Primary
WFL-09	Whale Ticket Dominance in Small Draws	Minor	Volatile Code	Open	Primary
WFL-10	Missing Tier Range Validation in setTicketTier	Minor	Volatile Code	Open	AuditBase ★
WFL-11	No SPDX License Identifier	Minor	Coding Style	Open	Primary
WFL-12	High VRF Confirmation Count Adds Latency	Informational	Coding Issue	Open	Primary
WFL-13	No Pause / Circuit Breaker Mechanism	Informational	Coding Issue	Open	Primary
WFL-14	Supra Oracle Third-Party Trust Assumption	Informational	Coding Issue	Open	Primary
WFL-15	Missing NatSpec Documentation	Informational	Coding Style	Open	Primary

▲ = Reclassified in v2.1/v2.2 | ★ = New finding from external tool review | Verified = Confirmed on-chain

6. Findings

WFL-01 Last-Resort Blockhash Fallback — Conditional Trust Assumption [AMENDED v2.1/v2.2]

Medium ▲

Category	Severity	Location	Status
Volatile Code	Medium	`WindfallLotto.sol — fallback randomness path`	Open

▲ **AMENDED (v2.1):** Originally classified as Major. Downgraded to Medium following client response. The WindfallLotto team correctly noted that the blockhash path is a tertiary fallback requiring both Chainlink VRF and Supra Oracle to fail independently before it is reached, and that the fallback exists to protect draw liveness rather than as a design preference for weak randomness. CryptoAudit.PRO accepts that Major overstated the probability; however Medium is retained because severity reflects the consequence of exploitation (jackpot bias) independently of how many preconditions must align. The liveness tradeoff is acknowledged in the updated description.

Description

The contract implements a three-tier randomness fallback chain: Chainlink VRF V2 Plus (primary) → Supra Oracle (secondary) → blockhash (tertiary). The blockhash path is only reachable after both primary and secondary randomness mechanisms fail or time out, making it a last-resort liveness mechanism rather than the default draw resolution path. When this path is active, the contract reads a past blockhash after a configured BLOCKHASH_DELAY. On Polygon PoS, validators have non-trivial influence over block production. A sophisticated validator or colluding validator set could theoretically withhold or re-order blocks during the fallback window to skew the outcome. The DrawBlockhashArmed event is emitted when this path is armed, making it observable on-chain. As jackpot size grows, the financial incentive to engineer such a scenario increases proportionally. The fallback does provide an important liveness benefit: without it, a draw could remain permanently unresolved if both oracles fail.

Recommendation

For future versions, consider removing the blockhash fallback and replacing it with retryable VRF or Supra requests after the timeout window. If blockhash is retained, commit to multiple future blockhashes across separate blocks (XOR'd together) to reduce any single validator's influence. Disclose clearly in the user interface when the blockhash fallback has been triggered for a given draw, so participants are aware of the reduced randomness guarantee for that specific draw.

WFL-02 Operational Reliance on Host Execution [AMENDED v2.1/v2.2]

Informational ▲

Category	Severity	Location	Status
Operational Risk	**Informational**	`closeDrawAndRequestRandom · processTickets · finalizeTier · openNextDraw`	**Verified**

▲ **AMENDED (v2.1):** Originally classified as Centralization. Reclassified to Informational following client response. The WindfallLotto team correctly identified that on-chain transaction history shows who has called lifecycle functions, not who is permitted to call them. If the draw lifecycle functions carry no onlyOwner or equivalent access restriction, the original "exclusive owner control" characterisation was imprecise. This finding is reclassified pending verification against the next draw close. CryptoAudit.PRO retains an operational dependency note: permissionless lifecycle functions place a gas-cost and execution burden on the community in the host's absence, which is a practical dependency even without a technical restriction.

Description

The draw lifecycle functions (closeDrawAndRequestRandom, processTickets, finalizeTier, openNextDraw) appear to be publicly callable and not exclusively restricted to the contract owner. Historical transaction activity shows that the host/operator address (windfall-lotto.eth) has been the only party actively performing these operations to date. This reflects operational practice rather than exclusive technical access control. The practical risk is operational reliance on the host or community to pay gas and execute draw maintenance transactions in a timely manner. If the host is unavailable and no community member steps in, draw progression may stall even without any technical access restriction. Status: Pending verification against next draw close.

Recommendation

Consider implementing a dedicated draw keeper role, automated keeper service (e.g., Chainlink Automation), or an explicit public incentive mechanism to ensure timely draw lifecycle execution. Document the open-call model clearly in public materials so that community members understand they can and may need to call these functions independently. Publish a public security and operations policy describing the expected operational cadence.

WFL-03 Operational Key Separation Recommendation — Treasury and Oracle Wallet [AMENDED v2.1/v2.2]

Informational ▲

Category	Severity	Location	Status
Operational Risk	Informational	`Constructor args [4] _hostTreasury and [8] supraWalletAddress`	Open

▲ **AMENDED (v2.1):** Originally classified as Centralization. Reclassified to Informational following client response. The team correctly noted that compromise of the shared address does not provide a direct mechanism to withdraw the jackpot, alter winning tickets, blacklist users, or change ticket outcomes outside contract rules. The risk is operational key management concentration, not direct protocol fund compromise. The team has also committed to migrate to a multi-signature wallet in a future deployment, which is the appropriate remediation.

Description

The current deployment uses the same address (0x0d18342127412658AEf47bA799C27259c4C17268) for both the host treasury (_hostTreasury, constructor arg [4]) and the Supra oracle wallet (supraWalletAddress, constructor arg [8]). This creates operational key management concentration: a single private key controls both protocol fee accumulation and oracle funding operations. Compromise of this address does not provide a direct function to withdraw the jackpot, manipulate randomness outcomes, blacklist users, or change ticket results. The risk is limited to operational availability (inability to fund oracle requests) and potential loss of accumulated host fees. The team plans to migrate to a multi-signature wallet in a future deployment and may separate the oracle funding wallet from the treasury address.

Recommendation

Use separate dedicated addresses for the host treasury and the Supra oracle funding wallet. The oracle wallet requires hot key access for top-up automation; the treasury should be held in cold storage or a multi-signature wallet. Implement the planned multisig migration and separate the oracle funding address from the fee-accumulating treasury address in the next contract version or parameter update.

WFL-04 No Emergency Fund Recovery Mechanism

Medium

Category	Severity	Location	Status
Volatile Code	Medium	`WindfallLotto.sol — no emergencyWithdraw`	Open

Description

The contract holds user DAI accumulated from ticket sales and donations. No emergencyWithdraw, pause, or fund-rescue function exists in the ABI. If a critical bug causes a draw to become permanently stuck (all three randomness sources fail simultaneously, or a logic error prevents state progression), jackpot DAI cannot be returned to participants or recovered by the owner.

Recommendation

Implement a time-locked emergency withdrawal allowing the owner to recover funds after an extended period of inactivity (e.g., 30 days with no successful draw close). Integrate OpenZeppelin Pausable to halt ticket sales and donations if a critical issue is detected. Emit a public event when emergency mode is activated.

WFL-05 Ticket Price Is Immutable — No Market Peg Adjustment — Medium

Category	Severity	Location	Status
Volatile Code	Medium	`TICKET_PRICE = 1e18 (1 DAI)`	Open

Description

TICKET_PRICE is hardcoded as 1 DAI (1 × 10^18 wei) and set immutably at construction. While DAI targets $1 USD, the protocol has no mechanism to respond to persistent de-pegging, change the price for market health, or adjust for inflationary conditions over the protocol's lifetime. There is also no mechanism to change the payment token itself.

Recommendation

Make TICKET_PRICE an owner-adjustable parameter protected by reasonable min/max bounds and a timelock delay, ensuring price changes are transparent. This enables the protocol to respond to economic conditions without redeployment.

WFL-06 BPS Values Lack Constructor Sum Invariant Check — Medium

Category	Severity	Location	Status
Coding Issue	Medium	`Constructor — BPS configuration`	Open

Description

The protocol uses multiple BPS share constants (HOST_FEE_BPS, TIER3_SHARE_BPS, TIER4_SHARE_BPS, TIER5_SHARE_BPS, MINTER_ROYALTY_BPS) that together govern fund distribution. If these values exceed 10,000 BPS in sum on any future redeployment, prize payouts would be underfunded or the contract would revert during distribution. No runtime check validates that the sum is within bounds at construction.

Recommendation

Add a constructor assertion: require(HOST_FEE_BPS + TIER3_SHARE_BPS + TIER4_SHARE_BPS + TIER5_SHARE_BPS + MINTER_ROYALTY_BPS <= 10000, "BPS overflow"). This is a cheap one-time check that prevents misconfiguration on any future deployment.

WFL-07 VRF Funding Dependency Before Fallback Activation — Operational Availability Risk [AMENDED v2.1/v2.2]

Minor ▲

Category	Severity	Location	Status
Volatile Code	Minor	`closeDrawAndRequestRandom — _requestChainlinkRandom call`	Open

▲ **AMENDED (v2.1):** Originally classified as Medium. Reclassified to Minor following client response. The team correctly identified that VRF subscription underfunding is recoverable — any external party can fund a Chainlink VRF subscription, not only the host operator. A temporary draw stall that can be resolved by anyone topping up the subscription is materially different from a permanent lockout. CryptoAudit.PRO retains the architectural design observation: the intent of having three randomness sources is undermined when the first source's failure blocks the second from activating. The try/catch recommendation remains unchanged.

Description

The closeDrawAndRequestRandom function calls _requestChainlinkRandom(currentDrawId) without a try/catch block. If the Chainlink VRF subscription has insufficient LINK tokens, the VRF coordinator reverts — and this revert propagates up, causing the entire closeDrawAndRequestRandom transaction to fail. This is a real operational scenario. When it occurs, the draw cannot be closed, even though Supra Oracle is configured as a backup. The full fallback chain is bypassed by this revert behaviour. The issue is recoverable: the Chainlink VRF subscription can be funded by any external party (community members, shareholders, or supporters), after which the draw can proceed normally. The finding represents an operational availability dependency, not a theft or manipulation vector.

Recommendation

Wrap the VRF request in a try/catch block. On VRF failure (e.g., insufficient subscription balance), automatically fall through to request randomness from the Supra oracle instead. Emit a VRFFallbackTriggered event so the operator is alerted. This makes the three-tier fallback chain function as intended — Supra activates automatically rather than requiring LINK top-up before any draw can proceed.

WFL-08 VRF Callback Gas Limit Is Excessively High

Minor

Category	Severity	Location	Status
Gas Optimization	Minor	`callbackGasLimit = 1,000,000`	Open

Description

The Chainlink VRF callback gas limit is set to 1,000,000 gas (constructor arg [5]). While lottery number resolution in the callback is unlikely to consume anywhere near this amount, Chainlink subscriptions are charged based on the configured gas allowance. Unused allowance translates to unnecessary LINK cost per VRF request over the protocol's lifetime.

Recommendation

Profile the actual gas consumed by rawFulfillRandomWords and set callbackGasLimit to approximately 150% of the measured maximum. The owner can update this via setVRFCallbackGasLimit without redeployment. Recommend profiling and reducing to ~200,000–300,000.

WFL-09 Whale Ticket Dominance in Small Draws

Minor

Category	Severity	Location	Status
Volatile Code	Minor	`MAX_TICKETS_PER_BUY = 50 · buyTickets()`	Open

Description

A single wallet can purchase up to 50 tickets per transaction and there is no per-wallet cap across the draw period. During early-stage or low-participation draws, a single actor can capture a majority probability share across all prize tiers. The WINDFALL_TRIGGER minimum jackpot threshold helps ensure a minimum pool but does not bound individual ticket concentration.

Recommendation

Consider adding a per-wallet ticket limit per draw (a mapping tracking each address's ticket count per drawId), or increase the batch cap threshold only when total ticket count exceeds a minimum. This improves fairness for all participants.

WFL-10 Missing Tier Range Validation in setTicketTier [NEW ★]

Minor

Category	Severity	Location	Status
Volatile Code	Minor	`WindfallTicket.sol — setTicketTier()`	Open

Description

The setTicketTier function in WindfallTicket.sol accepts a uint8 tier parameter (range 0–255). Only values 0–5 are semantically meaningful: 0 = no win, 3–5 = prize tiers. The function is protected by onlyLotto access control, preventing public exploitation. However, if WindfallLotto's tier assignment logic were to compute an out-of-range value due to a future bug or contract upgrade, it would be stored without revert, silently corrupting the ticket's prize eligibility.

Recommendation

Add a range validation inside setTicketTier: require(tier <= 5, "Invalid tier"). This is a low-cost defensive guard that prevents silent state corruption and makes any upstream miscalculation immediately visible via a revert.

WFL-11 No SPDX License Identifier in Source Files

Minor

Category	Severity	Location	Status
Coding Style	Minor	`All .sol source files`	Open

Description

The contract is deployed with no license specified (shown as -NA- on Polygonscan). The Solidity compiler recommends including an SPDX-License-Identifier comment in all source files. Its absence generates compiler warnings and leaves intellectual property terms undefined.

Recommendation

Add // SPDX-License-Identifier: MIT (or the appropriate license) to the top of all source files.

WFL-12 High VRF Confirmation Count Adds Draw Latency — Informational

Category	Severity	Location	Status
Coding Issue	Informational	`requestConfirmations = 15`	Open

Description

The VRF request confirmation count is set to 15. On Polygon, blocks are produced every ~2 seconds, so 15 confirmations adds approximately 30 seconds of latency before randomness is delivered. Chainlink's Polygon recommendation is typically 3–5 confirmations.

Recommendation

Reduce vrfRequestConfirmations to 3–5 via the setVRFRequestConfirmations function for faster draw resolution.

WFL-13 No Pause / Circuit Breaker Mechanism — Informational

Category	Severity	Location	Status
Coding Issue	Informational	`WindfallLotto.sol — no Pausable`	Open

Description

There is no pause() / unpause() mechanism. If a vulnerability is discovered post-deployment, the protocol cannot halt ticket sales or donations while a fix is prepared. Users continue sending DAI into a potentially compromised contract.

Recommendation

Inherit OpenZeppelin's Pausable contract and gate buyTicket, buyTickets, and donateToJackpot behind the whenNotPaused modifier. Emit a public event when the pause is activated.

WFL-14 Supra Oracle Introduces Third-Party Trust Assumption — Informational

Category	Severity	Location	Status
Coding Issue	Informational	`SUPRA_ROUTER · supraCallback()`	Open

Description

The contract trusts the Supra router to call supraCallback with the correct nonce and RNG values. While nonce mapping validates the call, the integrity of the randomness itself depends on Supra's internal security model. Unlike Chainlink VRF, Supra's callback does not provide an on-chain cryptographic proof that users can independently verify.

Recommendation

Document the Supra trust assumption clearly in public-facing materials. Investigate whether Supra's verifiable randomness proof system is available on Polygon and whether it can provide user-verifiable guarantees.

WFL-15 Missing NatSpec Documentation on Public Functions — Informational

Category	Severity	Location	Status
Coding Style	**Informational**	`All public and external functions`	**Open**

Description

The contract exposes 30+ public functions and events but provides no NatSpec (@notice, @param, @return) documentation. This makes integration harder for front-end developers, third-party tooling, and future auditors.

Recommendation

Add NatSpec comments to all public and external functions, especially those handling user funds (buyTicket, claim, donateToJackpot).

7. External Tool Review

Two external automated tools were submitted for cross-reference: SolidityScan (22 checks) and AuditBase AI (6 findings). Each finding was independently evaluated against the deployed bytecode and ABI before any determination was made. This section is unchanged from v2.0.

7.1 SolidityScan — Automated Threat Analysis

SolidityScan performed an automated scan of 22 checks. Important note: SolidityScan is primarily designed for ERC-20 token contracts. WindfallLotto is a lottery protocol — not a token — so 9 of SolidityScan's checks are entirely irrelevant.

SolidityScan Finding	Claimed Result	Our Verdict	Notes
Solidity pragma version	Low Risk	**Partially Valid**	v0.8.31 is recent; floating pragma concern if ^0.8.x used
Renounced ownership	Moderate Risk	**Valid — Confirmed**	Aligns with WFL-02; owner controls critical functions
Overpowered owners	No Impact	**FALSE NEGATIVE**	Owner controls all draw lifecycle — significant power missed
Critical admin functions	No Impact	**FALSE NEGATIVE**	Draw lifecycle functions are critical but not detected
External call risk in critical fns	No Impact	**FALSE NEGATIVE**	buyTicket and claim both make ERC-20 external calls
Hardcoded addresses	No Impact	**FALSE NEGATIVE**	Immutable constructor-set addresses not detected
9x ERC-20 specific checks	Various pass	**IRRELEVANT**	Tool not suited to lottery contract architecture
No self-destruct, no blacklist	Beneficial	**Confirmed Valid**	These checks do correctly pass for this contract

7.2 AuditBase — AI-Generated Findings

IMPORTANT — AI Hallucination Detected: AuditBase's report contains the function name _setFermentedJars in Finding #2. This function does not exist anywhere in the WindfallLotto codebase. It is a fabricated name generated by the AI model. No AI-generated finding should be accepted into an audit report without independent verification against actual bytecode or source code.

AuditBase Finding	Claimed Severity	Verdict	Action
Reentrancy in fulfillRandomWords	**HIGH**	**False Positive**	Coordinator-only access; not re-entrant; dismissed
Gas reversion (_setFermentedJars)	**MEDIUM**	**Hallucinated Function Name**	Function does not exist; gas concern captured in WFL-08
Unchecked call in closeDrawAndRequestRandom	**MEDIUM**	**VALID — New Finding**	Incorporated as WFL-07 after independent validation
Insufficient input validation setTicketTier	**MEDIUM**	**Partially Valid**	Severity reduced to Minor; incorporated as WFL-10
Revert in distributeHostFee	**MEDIUM**	**Partially Valid**	Low risk with SafeERC20; added as informational note
Reentrancy in claim — CRITICAL	**CRITICAL**	**False Positive**	nonReentrant modifier present; AI cited it in its own snippet

8. Positive Security Observations

✓	**MetaMask Delegation Toolkit Integration**	The draw lifecycle uses EIP-7710 delegation framework for closeDrawAndRequestRandom: 0x46c5642c pre-signed a delegation allowing a separate relayer key to submit the transaction without holding the principal private key. This is a mature key-separation pattern that reduces operational risk for the draw-closing step and demonstrates thoughtful operational security design beyond basic EOA usage.
✓	**Reentrancy Protection**	OpenZeppelin ReentrancyGuard applied on all user fund-handling functions. AuditBase's CRITICAL reentrancy claim is directly refuted by the confirmed nonReentrant modifier.
✓	**SafeERC20 Usage**	All DAI transfers use SafeERC20, protecting against non-standard ERC20 return value behaviour and failed transfers.
✓	**7 Zero-Address Guards**	Dedicated custom errors validate every address at construction: ZeroToken, ZeroTreasury, ZeroDrawNFT, ZeroFeeShare, ZeroTicket, ZeroSupraRouter, ZeroSupraWallet.
✓	**Chainlink VRF V2 Plus**	Best-in-class on-chain verifiable randomness as primary source, with cryptographic proofs independently verifiable by any party.
✓	**Two-Step Ownership Transfer**	Chainlink's ConfirmedOwner pattern prevents accidental ownership loss — both transferOwnership and acceptOwnership must be called.
✓	**No Self-Destruct / No Blacklisting**	Contract cannot be destroyed, users cannot be blocked, no hidden admin roles — confirmed by both primary analysis and SolidityScan.
✓	**35+ Custom Error Types**	Custom Solidity errors used throughout, reducing gas costs for failed transactions and improving debugging.
✓	**Draw NFT Provenance**	Two NFTs minted per draw (host + contract) provide immutable on-chain proof of each draw's occurrence.

9. Remediation Priority

The following table ranks all open findings by recommended remediation priority, balancing implementation effort against security impact. Priorities reflect the amended v2.1 severity classifications.

Priority	ID	Action	Effort	Impact	Severity
1	**WFL-01**	Remove or harden blockhash randomness fallback (retryable VRF/Supra)	Medium	High — prevents randomness bias	**Medium**
2	**WFL-04**	Add emergency withdrawal with 30-day timelock	Medium	High — user fund protection	**Medium**
3	**WFL-07**	Wrap VRF request in try/catch with Supra auto-fallback	Low	Medium — prevents draw stall	**Minor**
4	**WFL-03**	Separate host treasury from Supra wallet; implement multisig	Low	Medium — reduces key risk	**Informational**
5	**WFL-06**	Add BPS sum invariant check to constructor	Very Low	Medium — deployment guard	**Medium**
6	**WFL-13**	Add Pausable circuit breaker	Low	Medium — incident response	**Informational**

7	**WFL-02**	Verify permissionless access model; document community keeper path	Low	Low — operational clarity	**Informational**
8	**WFL-10**	Add require(tier <= 5) in setTicketTier	Very Low	Low — defensive guard	**Minor**
9	**WFL-05**	Make ticket price adjustable with timelock	Medium	Low-Medium — flexibility	**Medium**
10	**WFL-09**	Add per-wallet ticket cap per draw	Low	Low — fairness	**Minor**
11	**WFL-08**	Reduce VRF callback gas after profiling	Very Low	Low — cost savings	**Minor**
12	**WFL-12**	Reduce VRF confirmation count to 3–5	Very Low	Low — latency reduction	**Informational**
13	**WFL-11**	Add SPDX license identifiers	Very Low	Low — compliance	**Minor**
14	**WFL-14**	Document Supra trust assumption publicly	Very Low	Low — transparency	**Informational**
15	**WFL-15**	Add NatSpec to all public functions	Low	Low — developer experience	**Informational**

10. Optimizations

The following optimizations do not represent security vulnerabilities but are recommended to reduce gas costs, improve performance, and enhance developer experience.

ID	Optimization	Category	Description
OPT-01	Reduce VRF callback gas limit	Gas	Profile rawFulfillRandomWords and reduce callbackGasLimit from 1,000,000 to approximately 150% of measured maximum (estimated 200,000–300,000). Owner can update via setVRFCallbackGasLimit.
OPT-02	Lower VRF confirmations to 3–5	Performance	Reduce requestConfirmations from 15 to 3–5. On Polygon PoS, 15 blocks (~30s) adds unnecessary latency. Apply same reduction to supraConfirmations and blockConfirmations.
OPT-03	Pack draw struct fields	Gas	The Draw struct contains many uint types. Review slot alignment and pack smaller uints together (e.g., uint32, uint8) to reduce SLOAD/SSTORE costs per draw lifecycle transaction.
OPT-04	Use immutable for constructor-set constants	Gas	State variables set only in the constructor and never changed should be marked immutable, avoiding SLOAD costs and saving ~2,100 gas per read.
OPT-05	Add NatSpec for auto-documentation	Developer Experience	Adding @notice, @param, @return comments enables auto-generated documentation and improves Polygonscan and IDE integration for integrators and community reviewers.

Appendix — Finding Categories

Major	Logical errors that under specific circumstances could result in fund losses or loss of project control.
Medium	Issues that may not pose a direct risk to funds but affect overall functioning or reliability.
Minor	Smaller-scale issues that generally do not compromise overall integrity but may create future risk vectors.
Informational	Recommendations to improve code style, documentation, or adopt industry best practices.
Centralization	Design choices that designate privileged roles or create centralized control vectors over critical operations.

11. Disclaimer

This security assessment report was prepared by **CryptoAudit.PRO** for the WindfallLotto project. The assessment was conducted using bytecode analysis, ABI inspection, constructor argument decoding, on-chain transaction review, and evaluation of third-party automated scanning outputs. The assessment did not include access to the complete Solidity source files for all sub-contracts (WindfallTicket, WindfallDrawNFT, WindfallSVG) beyond what was accessible via the Polygonscan file browser and ABI.

This report is not, nor should it be considered, an endorsement or disapproval of any particular project or team. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed.

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. Blockchain technology and cryptographic assets present a high level of ongoing risk.

All findings were open and unresolved at original assessment (May 5, 2026). This report is version v2.2 — Final, incorporating v2.1 reclassifications (client response) and v2.2 on-chain verification of WFL-02 (May 8, 2026). A follow-up re-audit is recommended after remediation.

FOR AVOIDANCE OF DOUBT, THIS REPORT SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

CryptoAudit.PRO — Investigations & Audits
WindfallLotto Security Assessment | Polygon Mainnet | v2.2 Final | May 2026

cryptoaudit.pro | windfall-lotto.com