SECURITY AUDIT

SMART CONTRACT SECURITY REPORT — SHERLOCK

SpaceX100 ($SPX) · Ethereum Mainnet
Auditor: dEEP — Sherlock Private Security Researcher
Result: PASSED · No critical or high severity findings.

AUDIT SUMMARY

PROJECT	SpaceX100 — $SPX Token Ecosystem
AUDITOR	dEEP (Sherlock Private Security Programme)
AUDIT TYPE	Manual review + Slither static analysis + Mythril symbolic execution
SCOPE	Presale.sol | PresaleToken.sol | TeamVesting.sol | LiquidityLock.sol
CRITICAL FINDINGS	0 — None identified
HIGH FINDINGS	0 — None identified
MEDIUM FINDINGS	1 — Informational, non-exploitable in current state
LOW FINDINGS	3 — Minor best-practice deviations
INFORMATIONAL	2 — Operational observations
AUDIT DATE	June 2026
OVERALL RESULT	**PASSED — Cleared for mainnet deployment**

VERDICT: PASSED

No critical or high severity issues. The SpaceX100 contract suite is cleared for mainnet deployment.

01 EXECUTIVE SUMMARY

dEEP was commissioned to conduct a comprehensive security audit of the SpaceX100 smart contract suite prior to mainnet deployment. The audit covered four contracts: Presale.sol (UUPS upgradeable presale), PresaleToken.sol (ERC-20 token), TeamVesting.sol, and LiquidityLock.sol.

The audit involved manual line-by-line code review, automated static analysis using Slither and Mythril, and adversarial scenario modelling including re-entrancy attacks, oracle manipulation, front-running, privilege escalation, and economic attack modelling.

No critical or high severity vulnerabilities were identified. The codebase demonstrates strong security hygiene: OpenZeppelin ReentrancyGuard, SafeERC20, and Chainlink oracle staleness validation are correctly implemented throughout.

> Verdict: PASSED — No critical or high findings. One medium (informational) and three low findings disclosed and acknowledged.

02 AUDIT SCOPE & METHODOLOGY

Contracts in Scope

Presale.sol

UUPS upgradeable main presale contract. Controls purchases, stage management, bonus distribution, blacklist/whitelist, and treasury routing. Primary audit focus.

PresaleToken.sol

ERC-20 token (OZ ERC20Burnable). Handles initial 10B token distribution to 8 wallets. Includes transfer lock mechanism.

TeamVesting.sol

Holds 1,000,000,000 SPX (10%). 2-year cliff vesting. release() callable by owner after cliff.

LiquidityLock.sol

Holds 2,000,000,000 SPX (20%). 365-day time-lock. unlock() callable by owner after expiry.

Methodology

- Manual line-by-line code review of all in-scope contracts
- Slither static analyser — full detector suite (re-entrancy, access control, overflow)
- Mythril symbolic execution — arbitrary jumps, unprotected state changes
- Business logic review — token economics, presale mechanics, bonus accounting
- Oracle security — Chainlink integration, staleness handling, manipulation scenarios
- Adversarial scenarios — front-running, griefing, privilege escalation, DoS

03 FINDINGS OVERVIEW

Severity Classification

CRITICAL	Direct loss of funds or complete contract compromise.
HIGH	Significant risk to funds or core functionality.
MEDIUM	Non-exploitable but represents a code quality concern.
LOW	Minor issue or best-practice deviation.
INFO	Informational observation — no security impact.

All Findings

ID	Severity	Title	Status
SPX-M01	MEDIUM	Incorrect parent initializer call order	Acknowledged
SPX-L01	LOW	Missing zero-address in adminTransferTokens	Fixed
SPX-L02	LOW	previewPurchaseETH reverts on stale oracle	Acknowledged
SPX-L03	LOW	Event emitted before state fully settled	Acknowledged
SPX-I01	INFO	PRICE_STALENESS_THRESHOLD not configurable	Acknowledged
SPX-I02	INFO	No event on blacklisted purchase attempt	Acknowledged

SPX-M01 Incorrect Parent Initializer Call Order

MEDIUM

Acknowledged

See detailed description below

CONTRACT	`Presale.sol`
FUNCTION	`initialize()`

Description

The Presale contract calls parent initializers as: __Ownable_init(), __Pausable_init(), __ReentrancyGuard_init(). OpenZeppelin's C3 linearisation for the inheritance chain (Initializable, UUPSUpgradeable, PausableUpgradeable, OwnableUpgradeable, ReentrancyGuardUpgradeable) expects PausableUpgradeable to be initialised before OwnableUpgradeable.

The Hardhat OpenZeppelin upgrade plugin emits a warning about this ordering. The current ordering is non-exploitable because all parent init functions are independent and idempotent in the current implementation.

Impact

No exploitable impact in the current deployment. The warning may confuse future developers during upgrade cycles and could potentially mask genuine initialisation bugs introduced in future versions.

Recommendation

Reorder calls to match C3 linearisation: __Pausable_init() before __Ownable_init(). Apply in the next UUPS upgrade and verify with the OpenZeppelin upgrade safety checker.

Status

Acknowledged by the SpaceX100 team. Scheduled for remediation in the next contract upgrade cycle. Current deployment is not at risk.

SPX-L01 Missing Zero-Address Validation in adminTransferToke

LOW

Fixed

See detailed description below

CONTRACT	`Presale.sol`
FUNCTION	`adminTransferTokens(address to, uint256 amount)`

Description

The adminTransferTokens() function includes a blacklist check but did not explicitly validate that the 'to' address is not address(0). While OpenZeppelin's ERC-20 transfer() would revert on a transfer to address(0), the error originates from the token contract rather than the presale contract, making debugging less clear.

Impact

Low. No token loss risk — OZ ERC-20 rejects address(0) transfers. The issue was one of debugging clarity and defence-in-depth. The explicit check was absent in the reviewed version.

Recommendation

Add: require(to != address(0), 'Presale: zero recipient'); at the top of adminTransferTokens(). This was subsequently confirmed present in the final production deployment.

Status

Fixed — Confirmed present in the production mainnet deployment code.

SPX-L02 **previewPurchaseETH Reverts When Oracle Is Stale** Acknowledged

See detailed description below

LOW

CONTRACT	`Presale.sol`
FUNCTION	`previewPurchaseETH(uint256 ethAmount)`

Description

The previewPurchaseETH() view function calls _getETHValueInUSD() internally, which reverts if the Chainlink price feed is stale. While correct for actual purchases, a view function reverting causes poor UX for frontends polling it to display estimated token amounts to users browsing the presale.

Impact

Low. No funds at risk. Frontend applications receive an unhelpful revert during oracle downtime rather than a graceful fallback, potentially causing blank UI states during Chainlink maintenance windows.

Recommendation

Add a try/catch wrapper in previewPurchaseETH() returning (0, 0, 0) with a bool staleFlag when the oracle check fails, rather than propagating the revert to callers.

Status

Acknowledged. The SpaceX100 frontend handles this gracefully with client-side error boundaries and fallback displays.

SPX-L03 Event Emitted Before Stage State Fully Settled

LOW

Acknowledged

See detailed description below

CONTRACT	`Presale.sol`
FUNCTION	`advanceStage()`

Description

In advanceStage(), the StageAdvanced event is emitted after stages[prev].active is set to false and currentStageId is incremented, but before stages[currentStageId].active is set to true. An off-chain event listener acting immediately on StageAdvanced could observe a transient state where no stage appears active.

Impact

Very low. EVM atomicity ensures no on-chain state inconsistency is possible within the transaction. Only off-chain listeners that make synchronous re-entrant contract reads immediately on event receipt are affected.

Recommendation

Move the emit StageAdvanced(prev, currentStageId) statement to after all state changes complete — specifically after stages[currentStageId].active = true. Zero gas impact.

Status

Acknowledged. Off-chain infrastructure confirmed to handle this correctly via delayed state reads after event receipt.

SPX-I01 INFO **PRICE_STALENESS_THRESHOLD Not Configurable** Acknowledged

See detailed description below

CONTRACT	`Presale.sol`
FUNCTION	`PRICE_STALENESS_THRESHOLD (constant: 3600)`

Description

The Chainlink oracle staleness threshold is hardcoded as a uint256 constant (3,600 seconds = 1 hour). Chainlink feed heartbeat intervals may vary during extreme market conditions. A configurable threshold allows adjustment without a full contract upgrade.

Impact

Informational. The current value of 3,600 seconds is appropriate for the ETH/USD Chainlink feed on Ethereum mainnet (heartbeat: 3,600s, deviation: 0.5%). No immediate security risk.

Recommendation

In a future upgrade, replace the constant with an owner-configurable state variable bounded by a maximum cap (e.g. 7,200 seconds) to prevent accidental over-relaxation.

Status

Acknowledged. Will be evaluated for inclusion in the next upgrade cycle.

SPX-I02 No Event Emitted on Blacklisted Purchase Attempt

INFO

See detailed description below

Acknowledged

CONTRACT	`Presale.sol`
FUNCTION	`notBlacklisted modifier`

Description

When a blacklisted address attempts to purchase, claim bonuses, or receive tokens, the transaction reverts silently with a string error. No event is emitted, making it impossible for monitoring systems to track blacklisted activity without parsing revert reasons.

Impact

Informational. No security impact. Reduces operational visibility for compliance and monitoring teams tracking blacklisted address activity.

Recommendation

Consider emitting a BlacklistedAttempt(address indexed user, bytes4 indexed selector) event before reverting. This requires converting the modifier to a function call pattern.

Status

Acknowledged. Team will evaluate for inclusion in the next upgrade.

04 INDIVIDUAL CONTRACT REVIEWS

PresaleToken.sol [PASSED]

`0xAf5E67b6dDc1EaF9cC65B381f5F60ED0896d9D89`

Standard OpenZeppelin ERC20Burnable. Transfer lock correctly implemented via _beforeTokenTransfer override. Initial 10B distribution to 8 addresses handled atomically in the constructor. Supply correctness verifiable on-chain. No findings.

TeamVesting.sol [PASSED]

`0x5B0FaC96c621EeD5C7453368426D5ca232C72d10`

Simple time-lock vesting. Cliff end correctly calculated at deployment (now + 2 years). release() protected by onlyOwner and post-cliff timestamp check. Balance checked before release to prevent over-distribution. No re-entrancy risk. No findings.

LiquidityLock.sol [PASSED]

`0x61478Ff4e579D226cCd476EBF1A1e4A91EF40522`

365-day token time-lock. unlock() restricted by timestamp check and onlyOwner. Emergency owner pattern is clean. No re-entrancy risk as only SPX (trusted ERC-20) is transferred. SPX cannot be rescued from this contract by design. No findings.

05 GENERAL RECOMMENDATIONS

The following are general best-practice recommendations not tied to a specific finding, representing operational security improvements for consideration post-deployment.

- Transfer ownership to a Gnosis Safe multisig wallet before mainnet launch. Single EOA ownership creates a single point of failure — key loss or compromise would be unrecoverable.
- Consider implementing a 48-hour timelock on sensitive admin functions (advanceStage, endPresale, setTreasury1, setTreasury2). This gives token holders time to observe and react to unexpected administrative actions.
- Consider a public bug bounty programme on Immunefi for ongoing security monitoring post-launch. Well-audited contracts still benefit from continuous external review and financial incentives.
- Document the UUPS upgrade process thoroughly. Upgrades require a re-audit of the new implementation and should only be executed from a multisig wallet.
- Monitor Chainlink ETH/USD oracle health using the Chainlink Data Feeds dashboard. Set up automated alerts for extended periods without price updates.

06 AUDITOR STATEMENT

This audit was conducted by dEEP, an independent security researcher participating in the Sherlock private auditor programme. The audit was performed in June 2026 on the SpaceX100 contract suite as described in the scope section.

This report reflects the security posture of the reviewed contracts at the time of the audit. It does not guarantee the complete absence of all vulnerabilities. Security audits are point-in-time assessments — subsequent code changes or ecosystem changes may introduce new risks.

dEEP and the Sherlock programme shall not be liable for any losses resulting from the use or misuse of the audited contracts. This report is provided for informational purposes only.

dEEP

Sherlock Private Security Researcher

June 2026

Full report available at spacex100.io/audit