

Ownbit

Multisig Security Audit

EVM V10, V11 and V21 Solana Multisig Program

A source-code security assessment with local execution evidence, explicit trust boundaries and actionable remediation guidance.

ENGAGEMENT	DETAIL
Prepared for	Ownbit
Produced by	GPT-6-Astra
Assessment date	26 October 2026 (Australia/Sydney)
Report version	1.0 - Initial assessment
Report ID	OWNBIT-SCA-2026-1026
Source baseline	1f8c78428726bf8059e39ff4e1cfe44cf5532f4b
Disposition	Open findings; no remediation reassessment performed

Assessment type: AI-assisted code review and security testing. This report is not a CertiK audit, certification, endorsement or assurance of the deployed system. CertiK-style report conventions inform presentation only.

The scope is the pinned public source snapshot. Production bytecode, deployment configuration and live custody operations were not established by this assessment.

Executive summary

The reviewed code implements three EVM authorization models and a Solana proposal executor. The EVM contracts bind spending signatures to the wallet, chain, payload and nonce; the Solana program binds approvals to stored instructions and validates typed accounts and its vault PDA. The reviewed tests did not identify a confirmed below-quorum spending bypass.

The principal actionable finding is an EVM availability weakness: a third-party relayer holding valid signatures can deliberately underfund execution gas, cause an inner call to fail and still consume the spending nonce. The original authorization then becomes unusable. This was reproduced for V10, V11 and V21, including the exact published V10-routing and V21 proxies in isolated local state.

CLASSIFICATION	COUNT	INTERPRETATION
Critical / High	0 / 0	None identified within this assessment; not proof of absence.
Medium	1	Gas-induced authorization consumption.
Low	2	Post-execution approval mutation and generic-call settlement semantics.
Informational advisories	5	Design, integration, deployment and build-assurance observations.

All three findings remain **Open**. The five advisories are reported separately and are not represented as unconditional exploits. In particular, current-policy evaluation of old signatures and permissionless execution of previously approved Solana proposals do not bypass a valid quorum.

What to address first

- Specify and enforce a signed execution-gas/failure policy before nonce consumption, then regression-test the local gas-grief reproduction.
- Freeze Solana approval state after execution and ensure transaction records distinguish call completion from token settlement.
- Publish reproducible build/deployment evidence, the intended V11 proxy variant, and explicit authorization-lifecycle and operational-policy documentation.

No repairs were applied to the audited source and no production transactions were sent. There is no client remediation response or fix verification in this version.

02 / ENGAGEMENT CONTROL

Scope and source integrity

The repository master branch was resolved through GitHub to the commit below. Each downloaded file was checked against its Git blob SHA-1; SHA-256 digests are preserved in the evidence manifest. No production source was edited.

```
https://github.com/bitbill/ownbit-multisig-contracts
Commit: 1f8c78428726bf8059e39ff4e1cfe44cf5532f4b
Commit authored: 17 September 2026
Assessment date: 26 October 2026
```

IN-SCOPE UNIT	LINES	ROLE
V10 implementation	255	Fixed-threshold EVM multisig
V11 implementation	333	Inactivity-adjusted EVM multisig
Shared V11 proxy	26	Shared proxy; active constant routes to V10
V21 implementation	448	Weighted signers; owner-governed configuration
V21 proxy	53	Immutable V21 delegation proxy
Solana lib.rs	280	Solana proposal creation, approval and execution

Total: 1,395 physical source lines across six code files, including comments and blank lines. README.md (46 lines) was reviewed as supporting documentation. These counts are not executable-code coverage metrics.

Excluded or not established

Older contract versions, client/server policy implementation, key storage, WalletConnect transport, wallet UI, deployment factories, production indexers, supported-token inventories and off-chain membership controls are outside the audited source scope. Deployed bytecode equivalence, live signer sets, Solana upgrade authority and every target chain were not verified. Public explorer lookups did not return usable verification data.

The public Solana directory contains only lib.rs. The build manifest, dependency lock, feature flags and deployment artifact were not supplied. Local dependency selections are explicitly reconstructed, not assumed to match production.

03 / SYSTEM MODEL

Architecture and trust boundaries

COMPONENT	AUTHORIZATION	EXECUTION / STATE
EVM V10	Distinct owners meet a fixed headcount threshold.	Signed target/value/data + wallet + chain + spendNonce; arbitrary CALL.
EVM V11	Same ownership model, with a 416-day inactivity policy.	Signing owners are refreshed before a second threshold check; ERC1271 uses a virtual refresh.
EVM V21	Weighted signers spend; distinct owners approve configuration.	Separate spend/config nonces; owner quorum can replace signers and weights.
Solana	Owners approve a stored proposal; anyone may execute after quorum.	Stored program, metas and data are invoked using the multisig vault PDA; did_execute prevents reuse.

Authority and trust assumptions

- **V21 owners ultimately control funds.** An owner quorum can appoint new signers, including itself. Separation of day-to-day signers does not remove this administrative authority.
- **V11 inactivity is intentional policy.** If owners stop sending on-chain activity transactions, the effective quorum can fall. Off-chain/ERC1271 message signing alone does not refresh activity.
- **Approved arbitrary calls can be dangerous by design.** A valid quorum can authorize allowances, untrusted programs or destructive business actions. The audit does not treat authorized arbitrary execution itself as an exploit.
- **Application controls are not established as on-chain restrictions.** These EVM sources do not implement recipient allowlists, daily/monthly caps, a Security Admin role or an emergency pause. Any such controls require separate enforcement review.
- **Deployment matters.** EVM proxies use fixed implementation addresses and have no upgrade endpoint. Solana upgrade authority is a deployment property not derivable from the declared program ID.

Signatures and key custody

EVM participant checks use ecrecover and therefore do not implement contract-signer/ERC1271 validation for owners themselves. The wallet can validate its own aggregate ERC1271 signatures. Client onboarding should prove control of configured keys and make this distinction clear.

04 / HOW CONCLUSIONS WERE REACHED

Methodology and evidence levels

The review combined line-by-line inspection, independent component workstreams, cross-component threat analysis, compiler checks and isolated adversarial tests. Candidate findings were challenged for exploit prerequisites and intended behavior before inclusion. No professional affiliation, certification or exhaustive formal verification is claimed.

METHOD	WORK PERFORMED	BOUNDARY
Source review	All six code files; storage, initialization, signature domains, quorum, role updates, external calls and lifecycle.	Human-readable code analysis assisted by AI; not a mathematical proof.
EVM execution	Unmodified sources compiled with solc 0.8.30, IR optimizer 200, Shanghai; local Ganache.	Selected build settings; not a production bytecode match.
Solana execution	Native account-guard tests plus SBF on a local validator, including actual SOL/SPL transfers.	Reconstructed dependencies/features; not production binary parity.
Policy enumeration	3,021,026 finite V11 policy comparisons for up to nine owners.	Independent model equivalence; not contract-level formal verification.
Independent challenge	Rejected initialization-takeover, duplicate-weight and generic arbitrary-call false positives.	No assertion that every possible attack was exhausted.

Evidence vocabulary

EVM runtime: the relevant code executed on a local EVM. **Solana runtime:** reconstructed SBF executed on a local validator. **Native handler:** exact Rust/Anchor validation executed, with CPI mocked. **Static:** derived from source. **Conditional:** depends on unpublished build or integration choices. A passing adversarial test may confirm a weakness, rather than indicate safe behavior.

Severity model

Critical: broad direct asset loss or fundamental control failure. **High:** substantial asset/control compromise with credible limited prerequisites. **Medium:** meaningful availability or authorization-lifecycle harm with stated prerequisites. **Low:** limited integrity, settlement interpretation or operational impact. **Informational:** design trade-offs, hardening and assurance gaps without a demonstrated unconditional exploit.

Severity is based on impact and exploitability in the reviewed context. No numeric safety score or guaranteed loss estimate is assigned.

05 / PRIORITIZED RESULTS

Findings register

ID	SEVERITY	ISSUE / OBSERVATION	EVIDENCE
OWN-01	Medium	Unsigned execution gas allows authorization griefing	Local EVM + proxy
OWN-02	Low	Executed Solana proposals allow subsequent approval-bitmap changes	Solana SBF + native
OWN-03	Low	Spent does not prove successful ERC20 settlement	Local EVM
OBS-01	Informational	Authorization lifetime remains a policy decision beyond quorum verification	Runtime + design review
OBS-02	Informational	Executed Solana proposal accounts retain their rent deposits	Solana SBF + native
OBS-03	Informational	Published sources do not establish reproducible deployment provenance	Build / source review
OBS-04	Informational	The published V11-named proxy selects the V10 implementation	Pinned proxy source
OBS-05	Informational	Legacy Anchor IDL handling creates a conditional preinitialization exposure	Conditional feature tests

Findings: OWN-01 to OWN-03. **Design and assurance advisories:** OBS-01 to OBS-05. All are open at this snapshot; no remediation was reviewed. Advisories do not imply a current-quorum bypass or confirmed production exploit.

OWN-03 is specifically an integration/settlement interpretation issue: generic CALL semantics can be intentional, while consumers treating Spent as token-settlement proof are unsafe. OBS-01 similarly records authorization lifetime choices without mislabeling current-policy validation as signature forgery.

OWN-01 / MEDIUM / AVAILABILITY / AUTHORIZATION GRIEFING

Unsigned execution gas allows authorization grieving

Components: EVM V10, EVM V11, EVM V21

Status: Open | **Evidence:** Reproduced in a local EVM using unmodified source; also reproduced through the submitted V10-selected and V21 proxies.

Source: [V10 implementation:85-118](#); [V11 implementation:97-129](#); [V21 implementation:131-186](#)

Description and prerequisites

All three spend functions allow anyone to relay valid signatures. The digest binds the wallet, destination, value, calldata, nonce and chain, but not execution gas or failure policy. After verification, spendNonce advances. A failed destination call emits SpendFailed without reverting the outer transaction. A relayer can therefore select an unsigned gas limit that causes a committed failure, invalidating signatures that would otherwise execute successfully. Ordinary failure-driven nonce advancement is documented; the issue is that an unrelated relayer can deliberately induce it.

Prerequisites: The relayer obtains valid, unconsumed signatures and wins transaction ordering. The target needs enough gas that it can fail while the wallet retains sufficient gas for its failure handling.

Scenario and evidence

1. Authorize the audit fixture GasTarget.work(4000). At a 3,000,000 gas limit it completes and emits Spent.
2. Restore the same snapshot. An unrelated relayer submits the identical authorization with a 500,000 gas limit.
3. The outer receipt succeeds, SpendFailed is emitted, target state is unchanged and spendNonce advances.
4. Retry the original signatures with sufficient gas: authorization is rejected because the nonce was consumed.

Validation: All three unchanged implementations reproduced the issue. The exact published V10-routing and V21 proxies also reproduced it using local implementation-code injection. Full gas measurements and results are in results/shared-tests.json; fixtures and reproductions are in tests/evm/.

Impact and remediation

A public-transaction observer can invalidate a legitimate payment or contract action, force new approvals and disrupt dependent queues without holding a signing key. The reproduction establishes availability harm, not theft or modification of the signed destination or amount.

Recommendation: Bind a reviewed execution-gas budget and failure policy into the signed intent. Reject insufficient outer gas before nonce consumption, accounting for EIP-150 and post-call overhead. Make deliberate queue skipping an explicitly authorized operation. Reliable/private submission and SpendFailed monitoring can reduce exposure but do not fix the on-chain weakness.

Limits: The target is an audit fixture. No mainnet attack or deployed-bytecode equivalence was established. Gas parameters are specific to the selected local compiler/EVM, not universal thresholds.

OWN-02 / LOW / AUDIT-TRAIL INTEGRITY

Executed Solana proposals allow subsequent approval-bitmap changes

Components: Solana multisig

Status: Open | **Evidence:** Reproduced in native tests and a local-validator SBF deployment of the unmodified Rust source.

Source: [Solana lib.rs:100-112](#); [Solana lib.rs:170-173](#); [Solana lib.rs:192-199](#); [Solana lib.rs:230-233](#)

Description and prerequisites

The approve handler verifies that the supplied owner belongs to the multisig, and its account constraints bind the transaction to that multisig. It then sets the corresponding signers entry to true without checking `did_execute`. Execution records completion by setting `did_execute`, but does not preserve a separate execution-time approval set. An owner who did not approve before execution can therefore change the proposal's current approval bitmap afterward. The account accurately records that this owner eventually approved the proposal, but it no longer independently identifies which owners authorized its execution.

Prerequisites: A proposal has executed using a sufficient subset of its owners. A different legitimate owner subsequently calls approve. A consumer interprets the current signers bitmap as the approval set that authorized execution.

Scenario and evidence

1. Create a three-owner, threshold-two multisig and execute a proposal approved by its first two owners.
2. Confirm the stored approval bitmap is `[true, true, false]` and `did_execute` is true.
3. Submit approve using the third owner: it succeeds and persists `[true, true, true]`, without another payment.
4. A current-state exporter cannot distinguish this late approval from the execution-time approval set without historical instruction data.

Validation: The local-validator test `approval_after_execution_changes_stored_signer_bitmap` reproduced the change after a real SOL CPI transfer. Native-entry tests independently reproduce it. Evidence: `tests/solana/runtime-results.json`, `runtime-tests.cjs` and `tests/native_entry.rs`.

Impact and remediation

Current-state audit exports and operational investigations can incorrectly attribute participation in an already completed payment. This can obscure which subset was responsible at execution time. The change does not permit a second execution, increase funds moved or bypass owner membership: `execute_transaction` continues to reject `did_execute` transactions, and only a legitimate owner can add its own bit. The severity is therefore Low and tied to audit consumers rather than asset custody.

Recommendation: Reject approve once `did_execute` is true, using the existing `AlreadyExecuted` error or an equivalent terminal-state check. If late acknowledgments are a desired product feature, store them separately from execution approvals. Consider emitting an execution event containing the approval bitmap or its hash. Consumers should reconstruct timing from historical instructions instead of treating the final account bitmap as an immutable execution record.

Limits: The local SBF binary uses reconstructed Anchor 0.30.1 and solana-program 1.18.26 dependencies, not a verified production build. No production indexer or Ownbit export was shown to make the incorrect inference described.

OWN-03 / LOW / INTEGRATION / EXECUTION-RESULT INTERPRETATION

Spent does not prove successful ERC20 settlement

Components: EVM V10, EVM V11, EVM V21

Status: Open | **Evidence:** Reproduced in local EVM execution of all three implementations with an audit-only false-returning token.

Source: [V10 implementation:98-118](#); [V11 implementation:110-129](#); [V21 implementation:152-186](#)

Description and prerequisites

The spending functions emit Spent whenever a low-level CALL returns success=true, ignoring successful-call return data. An ERC20 transfer may return false without reverting. Such a call consumes spendNonce and emits Spent although no tokens move. This is a distinction between generic execution and asset settlement; it is not a multisig or signature failure. Integrations must not interpret a call-completion event as proof of the requested payment.

Prerequisites: An authorized target returns application-level failure instead of reverting, and a client or accounting system treats Spent as settlement proof.

Scenario and evidence

1. Mint 1,000 units of FalseReturnToken to a test wallet. Its transfer function returns false without updating balances.
2. Authorize transfer(recipient, 100) through the normal quorum and submit with sufficient gas.
3. Observe outer success, Spent and a consumed nonce. Wallet balance remains 1,000; recipient balance remains zero.

Validation: The unchanged V10, V11 and V21 implementations emitted Spent for the audit token while balances remained unchanged. See [results/shared-tests.json](#), [tests/evm/shared-tests.cjs](#) and [fixtures.sol](#).

Impact and remediation

A payment may be reported as completed despite no asset movement, causing reconciliation errors or a missed payment. Only an already authorized call is affected; no unauthorized token movement is demonstrated. Treat this as a Low integration finding.

Recommendation: Define Spent as call-level completion. Reconcile token settlement using operation-aware return handling and authoritative token state/events. If adding reviewed ERC20 helpers or adapters, support standard true and compatible empty-return tokens while rejecting false. Do not decode every arbitrary call as bool: valid generic APIs can legitimately return other data.

Limits: The fixture does not establish behavior of every supported token or prove a production consumer currently misreports settlement. Generic CALL semantics may be intentional; the API and client contract must make them explicit.

OBS-01 / INFORMATIONAL / CONDITIONAL GOVERNANCE / LIFECYCLE ADVISORY

Authorization lifetime remains a policy decision beyond quorum verification

Components: EVM V21, Solana multisig

Status: Open | **Evidence:** Local EVM and Solana-validator behavior, supported by static lifecycle analysis; design advisory.

Source: V21 implementation:131-144; V21 implementation:293-310; V21 implementation:402-446; Solana lib.rs:100-131; Solana lib.rs:202-234

Description and prerequisites

V21 changes signer policy without advancing spendNonce or binding a signer-policy epoch into spending and ERC1271 digests. An old fragment from a retained signer can become sufficient after its weight increases. Solana approvals are monotonic, with no proposal expiry or cancellation transition; a quorum-approved proposal remains permissionlessly executable while its downstream instruction remains valid. These behaviors may be intentional. The supplied specification does not promise automatic revocation, so they are recorded as lifecycle considerations rather than unconditional vulnerabilities.

Prerequisites: An authentic pending authorization survives a later change in authority or business intent. No signature is forged and no valid quorum is bypassed.

Scenario and evidence

1. V21: retain a one-signer fragment from a two-weight quorum, then authorize a weight increase that makes the same signer sufficient. The unchanged fragment becomes usable.
2. Solana: approve a proposal to threshold, leave it unexecuted, and later invoke execution without a fresh owner signature.

Validation: V21 spending and ERC1271 state transitions were reproduced. The Solana native test reached its recording CPI boundary without fresh owner approval. The local Solana validator also executed a real quorum-approved SOL transfer without an owner signing execution.

Impact and remediation

A superseded intent may remain usable when users assume that a policy change or off-chain cancellation retired it.

Recommendation: Specify authorization lifetime clearly. If revocation is required, add an approval-bound epoch or expiry and a reviewed cancellation transition. Address cancellation/execution ordering explicitly.

Limits: Persistence is not a current-quorum bypass. Indefinite lifetime follows from the absence of an expiry/cancel gate, not a literal multi-year runtime experiment. Production deployment equivalence and intended business policy remain unverified.

OBS-02 / INFORMATIONAL / STORAGE LIFECYCLE / OPERATIONAL COST

Executed Solana proposal accounts retain their rent deposits

Components: Solana multisig

Status: Open | **Evidence:** Static review, native tests and an actual local-validator SOL transfer.

Source: [Solana lib.rs:170-173](#); [Solana lib.rs:210-234](#)

Description and prerequisites

The execution comment describes destroying the transaction, but the handler only sets `did_execute` to true. The proposal remains a program-owned account containing its instruction, approvals and completion flag. No close instruction or authenticated rent-refund path exists in the reviewed program. Keeping this state may be an intentional choice to preserve history and prevent re-execution; the comment and operational documentation should state that choice accurately.

Prerequisites: The wallet creates proposal state accounts and uses the submitted interface without a separate, verified closure mechanism.

Scenario and evidence

1. Allocate and initialize a funded proposal account, then execute an approved instruction.
2. Inspect the account after successful handler completion: its data and lamports remain present, with `did_execute` set.

Validation: After a real SOL transfer, the local-validator test retained the proposal data and 15,144,960 lamports in its 2,048-byte fixture account. Native execution independently confirms that only `did_execute` changes. This rent amount describes the test allocation, not a universal production fee.

Impact and remediation

Repeated use accumulates state accounts and rent deposits that users cannot reclaim through the current interface. This is an operating-cost and lifecycle issue, not an attacker-driven withdrawal of vault funds.

Recommendation: Document permanent retention, or design a closure flow with an authenticated refund recipient. Preserve replay prevention and historical evidence when accounts close; simply allowing an approved identity to be recreated would introduce a new risk.

Limits: Retention can be intentional history storage. The local SBF build is reconstructed; production rent costs, allocation sizes and usage rates were not measured.

OBS-03 / INFORMATIONAL / BUILD AND DEPLOYMENT ASSURANCE GAP

Published sources do not establish reproducible deployment provenance

Components: EVM V10, EVM V11, EVM V21, Solana multisig

Status: Open | **Evidence:** Pinned repository-tree inspection and recorded reconstruction builds.

Source: [V10 implementation:1](#); [V11 implementation:1](#); [V21 implementation:1-2](#); [V21 proxy:5-20](#); [Solana lib.rs:1-7](#)

Description and prerequisites

The repository supplies broad Solidity pragmas and hardcoded implementation addresses without a reproducible compiler configuration or deployment artifact attestation. The Solana subtree contains lib.rs without Cargo.toml, Cargo.lock, build features or SBF toolchain metadata. Audit builds therefore reconstruct explicit settings; they do not prove production parity. In particular, V21 failed default solc 0.8.30 code generation with Stack too deep and compiled successfully with vialR enabled.

Prerequisites: A deployment or user relies on the published source as evidence of the bytecode actually executing on-chain.

Scenario and evidence

1. Build the snapshot using independently selected dependencies and compiler settings.
2. Obtain executable test artifacts, but lack the original manifest and deployment evidence needed to identify them with production programs.

Validation: The audit records its solc 0.8.30 vialR/optimizer/Shanghai settings and reconstructed Anchor 0.30.1/Solana 1.18.26 dependencies. Source hashes are preserved separately. An audit-selected no-idl SBF build also succeeded and passed 12 local-validator tests, without establishing production binary parity.

Impact and remediation

Dependency-generated behavior, compiler choices and administrative deployment state remain outside the established assurance boundary.

Recommendation: Publish pinned manifests, lockfiles, build flags and commands, source/artifact hashes, chain-specific implementation mappings and reproducible verification instructions. For Solana include the deployed program-data metadata and upgrade-authority policy.

Limits: This is missing assurance evidence, not proof of a malicious deployment or vulnerable dependency. Production bytecode correspondence and upgrade authority were not verified.

OBS-04 / INFORMATIONAL / DEPLOYMENT CONFIGURATION CLARITY

The published V11-named proxy selects the V10 implementation

Components: EVM V10, EVM V11 proxy configuration

Status: Open | **Evidence:** Exact-source inspection and local constructor/delegation tests.

Source: [Shared V11 proxy:4-12](#); [README.md:3-9](#)

Description and prerequisites

OwnbitMultiSigV11(Proxy).sol assigns its active implementation constant to the address annotated as V10. The address annotated as V11 appears only in a commented declaration. The README describes using this proxy with either V10 or V11 implementations, but compiling the file unchanged selects only the active address. The filename therefore does not identify the resulting policy, and the exact source does not contain an executable V11-selected variant.

Prerequisites: A deployer treats the proxy filename or README pairing as selecting accident-protection V11 behavior without checking the active implementation constant.

Scenario and evidence

1. Compile the submitted proxy unchanged while intending to deploy accident-protection V11.
2. The constructor and fallback delegate to the V10-annotated constant instead of the commented V11 address.

Validation: Local tests compile the exact active proxy and verify its V10-bound delegation using injected implementation code.

Impact and remediation

A deployment process that omits this distinction can select the wrong wallet policy, including the absence of expected inactivity protection. Existing production misdeployment was not demonstrated.

Recommendation: Publish distinct, unambiguous deployment variants or generated per-chain artifacts with verified implementation addresses and code hashes. Assert expected version and policy behavior during deployment acceptance.

Limits: No source edit, commented-code activation or claim about the code currently deployed at either address is included in that test.

OBS-05 / INFORMATIONAL / CONDITIONAL DEPENDENCY / DEPLOYMENT ADVISORY

Legacy Anchor IDL handling creates a conditional preinitialization exposure

Components: Solana multisig, Reconstructed Anchor-generated dispatcher

Status: Open | **Evidence:** Conditional default-feature native reproduction; no-idl negative controls in native and local-validator tests.

Source: [Solana lib.rs:39-40](#); [Solana lib.rs:177-189](#)

Description and prerequisites

The application initialization accounts require their own signatures, but the reconstructed Anchor default build also exposes legacy IDL instructions. Its `IdlCreateBuffer` path accepts a zero-discriminator program-owned account without that account's signature. During a separate allocation/initialization window, another signer can claim the account as an IDL buffer before the intended handler initializes it. This does not affect already initialized multisig or proposal accounts. The public repository omits the dependency manifest and build features, so the condition is not established for production.

Prerequisites: Legacy generated IDL instructions are enabled, and a program-owned zero-discriminator account is allocated in a different transaction from its intended initialization.

Scenario and evidence

1. In the reconstructed default build, allocate a zeroed program-owned account and leave initialization pending.
2. Call the generated IDL buffer instruction with an unrelated authority and no signature from the allocated account; observe acceptance.
3. Repeat with the no-idl feature: the call is rejected and the account remains zeroed.

Validation: Both default and no-idl native tests exercise the generated dispatcher. The inspected iOS creation path combines allocation and initialization atomically. The reconstructed no-idl SBF program also rejected the legacy buffer operation on a local validator.

Impact and remediation

Affected split-transaction creation paths can lose the opportunity to initialize their newly allocated accounts as intended.

Recommendation: Keep allocation and initialization atomic. Publish and test actual build features; disable legacy IDL mutation when unnecessary.

Limits: This is a conditional build observation, not a confirmed deployed vulnerability. Other creation clients and the deployed SBF binary were not verified.

06 / VALIDATION SUMMARY

Execution evidence and coverage

SUITE	RESULT	WHAT THE RESULT ESTABLISHES
Shared EVM suite	66 / 66 passed	Domains, quorum rejection, replay, callbacks, same-nonce reentrancy, failure semantics and proxy execution.
V11 boundary suite	13 / 13 passed	416-day inclusive boundary, activity refresh and virtual-refresh behavior.
V21 governance suite	27 / 27 passed	Weighted subsets, configuration changes, owner/signer separation and old authorization behavior.
Solana SBF suite	12 / 12 passed	Real SOL/SPL CPI, quorum, replay, account/PDA binding, rollback/retry, late approvals.
Solana native suite	13 / 13, twice	Exact handler/account-guard behavior under default and no-idl builds; CPI mocked.
V11 finite policy model	3,021,026 matched	Acceptance agrees with the enumerated union-of-active-and-signing-owner policy.

Aggregate EVM result: 106 passing assertions across three suites. **Solana result:** 12 local-validator SBF tests plus 26 native test executions across two feature profiles (13 cases each). These totals include tests that intentionally reproduce reported issues.

Gas-induced failure: measured local executions

PATH	SUCCESS GAS USED	LOW-GAS USED	RESULT
V10 implementation	1,223,256	495,063	SpendFailed; nonce consumed
V11 implementation	1,237,211	495,369	SpendFailed; nonce consumed
V21 implementation	1,229,044	496,415	SpendFailed; nonce consumed
V10 exact proxy	1,243,178	488,147	SpendFailed; nonce consumed
V21 exact proxy	1,248,990	489,500	SpendFailed; nonce consumed

Control gas limit: 3,000,000. Relayer gas limit: 500,000. A deterministic audit-only loop target was used. Each comparison restored the same pre-execution snapshot and used the same signed intent; target state stayed unchanged after the low-gas attempt and original signatures were rejected afterward. Exact proxy tests injected compiled implementation code at hardcoded addresses only inside local Ganache.

NFT checks validated receiver callbacks and interface declarations, not complete ERC721/ERC1155 contract integrations. Solana SBF tests executed real local SOL and SPL Token transfers; native tests separately used CPI stubs. No live funds or production keys were used.

07 / COVERAGE INTERPRETATION

Security properties and remaining gaps

SURFACE	REVIEW RESULT	EVIDENCE
Initialization	Bounded/nonzero EVM quorums; unique participants; proxy constructor initialization and replay rejection.	Static + local EVM
Signature scope	Wallet, chain, payload and spending nonce are bound. ERC1271 has a distinct wrapped domain.	Static + local EVM
Quorum integrity	Repeated addresses do not multiply votes/weights; V21 removed signers are rejected.	Static + local EVM
V11 inactivity	Real and virtual reactivation use active/signing owner union; stale outsider cannot bypass active quorum.	EVM + finite model
Solana accounts	Typed ownership/discriminators, signer membership, canonical PDA and has_one relationships validated.	SBF + native + static
Solana instructions	Program, account metas and data are committed in proposal; ordered account identity is checked.	SBF + native + static
External execution	Nonce precedes EVM CALL; Solana marks execution after successful CPI. Runtime limitations remain.	EVM + local Solana SBF

Rejected or bounded hypotheses

- Initializing a standalone EVM implementation does not initialize proxy storage. No proxy takeover was demonstrated through this path.
- High-s spend signatures alone do not multiply quorum: recovered addresses are deduplicated and consumed nonces reject replay. Canonicalization remains useful hardening.
- V21 owner quorum replacing signers is an explicit administrative capability. It is not an unauthorized privilege escalation.
- Arbitrary Solana CPI requires approved stored instructions. The PDA constraint and runtime privilege rules cannot be replaced by a caller-chosen vault address.

Validation still required before a deployment assurance claim

Match production binaries to exact build inputs; inspect Solana upgrade authority; extend runtime coverage to Token-2022, privilege combinations and direct self-CPI scenarios; verify production receipt interpretation, V11 deployment selection and allocation atomicity. Review wallet/backend enforcement separately.

08 / ACTION PLAN

Remediation and release criteria

PRIORITY	ACTION	CLOSURE EVIDENCE
P1	Resolve OWN-01 with an explicit signed execution-gas and failure policy.	Same-intent low-gas reproduction cannot consume authorization unexpectedly; adequate-gas execution still succeeds.
P2	Reject Solana approve after execution; preserve execution-time participation.	Late approval fails; approved and executed state is unchanged; normal pre-execution approvals pass.
P2	Separate generic CALL completion from asset settlement.	False-return ERC20 scenario is correctly surfaced; arbitrary valid return values remain supported.
P2	Publish reproducible builds and deployment mapping.	Pinned compiler/dependencies/features, artifact hashes, bytecode/program comparison and authority records.
P3	Specify lifetime, cancellation, rent and integration behavior.	Documented choices plus regression cases for configuration changes, stale proposals and account closure if implemented.

Deployment-aware remediation

The submitted EVM proxies point to constant implementation addresses and expose no upgrade method. Source fixes do not change existing deployments. A release plan must identify affected deployments, new reviewed contracts, the authorization required to migrate assets, outstanding allowances and proposals, and a safe client transition. Do not describe a new source commit as remediation of existing wallets without migration evidence.

The Solana program may or may not retain an upgrade authority; this audit did not establish it. Confirm deployed ProgramData and authority policy before selecting an upgrade or migration path. Any change to account layout requires explicit state-versioning and migration tests.

Retest protocol

For each change, record the new commit, map it to finding IDs, rerun the original reproductions and relevant negative controls, review new attack surfaces and compare deployment artifacts. Mark a finding Resolved only after the relevant evidence is reviewed; otherwise use Open or a clearly documented design acceptance.

No team acknowledgment, accepted-risk decision or remediation completion is implied by this report.

APPENDIX A / AUDIT TRAIL

Reproduction and evidence package

The companion evidence archive contains exact source files, Git/SHA-256 manifests, compiler diagnostics, locked test dependencies, fixtures, result logs and this report generator. It excludes installed dependencies, build caches, vendor caches and local validator state.

EVM reproduction

```
cd tests/evm
npm ci --ignore-scripts --no-audit --no-fund
node shared-tests.cjs
node v11-specific.cjs
node v21-specific.cjs
```

Run from the extracted audit root as indicated. Node.js 20.20.2, solc 0.8.30, ethers 6.15.0 and Ganache 7.9.2 were used. Compiler settings: vialR=true, optimizer enabled with 200 runs, evmVersion=shanghai. V21 default non-IR code generation failed with Stack too deep; no source workaround was applied.

Solana native reproduction

```
cargo test --locked --manifest-path tests/solana/Cargo.toml \
  --test native_entry -- --test-threads=1
cargo test --locked --manifest-path tests/solana/Cargo.toml \
  --features no-idl --test native_entry -- --test-threads=1
```

Native reconstruction used Rust/Cargo 1.92.0, Anchor 0.30.1 and solana-program 1.18.26. Dependencies are audit-selected. The exact supplied lib.rs is referenced directly. Use --offline only if the required crates are already cached. CPI and rent sysvar stubs are visible in the harness.

Solana SBF reproduction

The local runtime used Solana validator 1.18.14 and SBF Rust 1.75.0, with Anchor 0.30.1 / solana-program 1.18.26 and no-idl. The runtime-project manifest and lockfile preserve this separate build. See tests/solana/README.txt for build, localhost-only validator and JavaScript commands.

Integrity checks

```
python3 verify_evidence.py
python3 review/evm10-11-policy-model.py
```

verify_evidence.py verifies source bytes against both Git blob IDs and SHA-256, and checks archived file hashes. Raw logs are the authoritative record of executed commands; narrative conclusions are scoped to the evidence labels used in this report.

APPENDIX B / IMMUTABLE BASELINE

Source fingerprint manifest

Repository commit: 1f8c78428726bf8059e39ff4e1cfe44cf5532f4b

README.md

Git blob: 91e9409a18ff8335fa12f4c606c25bdcc9fad6a8
SHA-256: 4ff81271734df6dd2e233521537cdf7f5c5386511d540cbac969f2a7e45fdd28
1790 bytes | 46 physical lines

OwnbitMultiSigV10(Implementation).sol

Git blob: 61c2ddf3a7c4f5a73c8572c33592b8c200c4380c
SHA-256: 11b6f7a8d0d528fee83f9dc9decd9b2b45160c2034fea391281ef4ca68707c13
11735 bytes | 255 physical lines

OwnbitMultiSigV11(Implementation).sol

Git blob: a0084ba3350083854cab981627a65ad746fa5916
SHA-256: 4f6635331e3a88ce621f5425335ccbd24e6146f4e7dd2b0d0619bce0c3472a1e
15090 bytes | 333 physical lines

OwnbitMultiSigV11(Proxy).sol

Git blob: 82a5866f0676e2786ffdf8a75fcb49a29f8a2d94
SHA-256: 7cab4a2821047ce428b3ffe292775d5205224e33298962485ec5f32eb873ac81
1068 bytes | 26 physical lines

OwnbitMultiSigV21(Implementation).sol

Git blob: 04e1c081edceb4e3220679cbef5cf66df7349473
SHA-256: f017d4c4c82eea48567a876a741ee37b639dfb67a40a206cd616a0b1435d6b57
19441 bytes | 448 physical lines

OwnbitMultiSigV21(Proxy).sol

Git blob: 6e990c04ad63efbd4f9954f417e51eadab2e77ec
SHA-256: 1a27bf59e0996dd907e8b69d2a7a659b0f5c68fee6d69827c9daf14e31deb878
2017 bytes | 53 physical lines

ownbit-solana-multisig/src/lib.rs

Git blob: 86a987195c7d67ed671850f74253b67981e2b369
SHA-256: 8cd58c34d47316683cd45289d199907d939f95648d50268660789461561c59ef
8715 bytes | 280 physical lines

The Git blob ID is computed over the canonical blob header plus file bytes. SHA-256 is computed over file bytes only. Line references throughout the report use this exact snapshot.

APPENDIX C / READER GUIDANCE

References and assessment limitations

- R1 Audited repository and pinned commit
- R2 ERC-1271: contract signature validation
- R3 EIP-150: call gas allocation
- R4 Solidity 0.8.30: low-level call semantics
- R5 Solidity security considerations
- R6 Solana CPI execution and privileges
- R7 Anchor account constraints
- R8 Anchor 0.30.1 generated IDL handlers
- R9 Anchor 0.30.1 IDL support
- R10 CertiK: published auditing methodology (format reference only)

Limits of reliance

This report assesses a particular public source snapshot using the methods and environments described. It does not establish that deployed contracts use the same bytecode or dependencies, that keys are securely held, that off-chain services enforce stated policies, or that all supported assets and chains behave identically. Absence of an identified critical/high finding is not a guarantee of security.

Native Solana tests replace the CPI boundary; separate SBF tests validate selected real SOL/SPL transfer and rollback paths. Neither environment establishes production build parity or covers every target program. Local EVM runtime and finite policy enumeration are not exhaustive formal verification. Compiler, dependency, chain, wallet, human-process and economic risks may remain outside the tested cases.

Produced by **GPT-6-Astra** as an AI-assisted assessment for Ownbit. No CertiK or other third-party auditor conducted, certified or endorsed this report. No remediation verification, continuous monitoring, legal compliance assessment or production incident investigation is included.