

Ownbit MultiSig Contracts — Security Assessment

Oct 26, 2026 · @Guopeng

Report Overview

Ownbit's EVM multisig contracts (V10, V11, V21) and its Solana multisig program were reviewed at commit `1f8c784`. The review identified 16 findings: none Critical, 3 Major, 4 Medium, 4 Minor and 5 Informational.

Item	Detail
Client	Ownbit (BITTOOL PTE. LTD.)
Project	Ownbit MultiSig smart contracts
Repository	bitbill/ownbit-multisig-contracts
Commit	<code>1f8c78428726bf8059e39ff4e1cfe44cf5532f4b</code> (2026-09-17)
Components	Standard MultiSig (V10), Accident Protection MultiSig (V11), Enterprise MultiSig (V21), Solana MultiSig program
Languages	Solidity $\geq 0.8.0 < 0.9.0$; Rust (Anchor)
Target chains	Ethereum, BNB Smart Chain, Base, Arbitrum One (per source comments); Solana mainnet, program <code>7uPTVnLrNM9bAdCGrfckz1ugJeRs5bEGcEgcwzPuFsAK</code>
Review type	Manual line-by-line review, differential review against V20, executable proof-of-concept tests
Prepared by	GPT-6-Astra (AI security review)
Report version	1.0, 2026-10-26

Disclosure. This assessment was produced by an AI security reviewer. It is not an audit by an accredited third-party security firm and should not be presented as one. Findings marked as confirmed were reproduced with executable tests; the absence of a finding is not proof that no vulnerability exists.

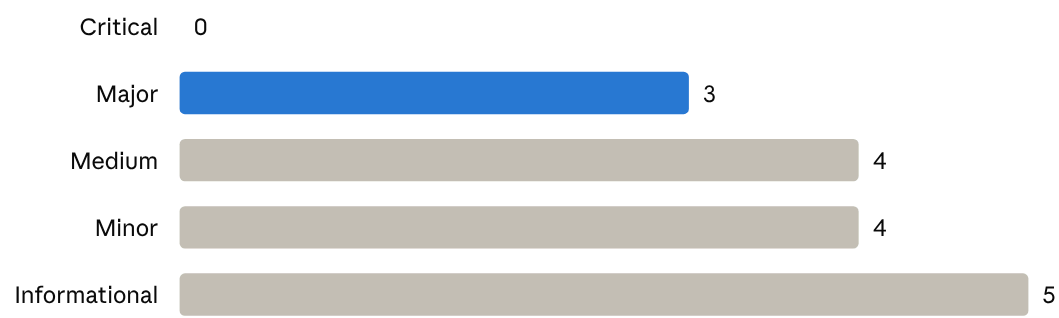
Executive Summary

The core signature logic is sound. The three Major findings are design or centralization risks: the 416-day dormancy rule in V11 (OMS-01), the Enterprise owners' power to replace signers (OMS-04) and the unverified Solana upgrade authority (OMS-05). The most urgent fix is rated Medium: the non-reverting `spend()` introduced in September 2026 (OMS-02) affects live wallets and needs only an app change. No finding lets an outsider holding no key move funds.

Replay protection is solid on every EVM contract. Spend digests bind the wallet address, nonce and live `block.chainid`; ERC-1271 digests use a separate EIP-712 domain, so a dApp signature cannot double as a spend vote. EVM proxies are immutable, with no admin or upgrade path.

3 of 16 findings are Major; none is Critical

16 findings by severity at commit 1f8c784



Findings Overview table in this report · 16 findings

Key risks

- **OMS-01** In an Accident Protection wallet, once the other owners are silent for 416 days, any one owner key can drain it, including through ERC-1271 permits.
- **OMS-02** (Medium) Depending on the node, `eth_estimateGas` can return a gas limit at which heavy calls fail silently and burn the nonce; a third party can force the same outcome.

- **OMS-03** (Medium) Solana approvals bind to a transaction account, not to content an offline signer can check, so a cold signer must trust the online device to show it the truth.
- **OMS-04** In Enterprise wallets, the owner quorum can replace all spending signers and spend in the same block.
- **OMS-05** The Solana program's upgrade authority and build reproducibility are unverified; an upgradeable program is a single point of control over every Solana vault.

Priority actions

1. Fix the client gas strategy now (OMS-02); this needs no contract change and prevents silent failures today.
2. Add a keep-alive reminder and auto- `active()` flow in the app for every Accident Protection wallet (OMS-01).
3. Confirm the Solana upgrade authority and publish a verifiable build (OMS-05).
4. In the next contract versions, add a signed gas floor, a challenge window for degraded-threshold spends, EIP-712 spend messages and content-hash approvals on Solana.
5. Commission an independent third-party audit of the next versions before deployment.

Audit Scope

Six source files, 1,395 lines in total, were reviewed at commit `1f8c784`. Hashes identify the exact files reviewed.

File	Component	Lines	SHA-256
<code>OwnbitMultiSigV10(Implementation).sol</code>	Standard MultiSig implementation	255	<code>11b6f7a8d0d528fee83f9dc9dec9b2b45160c2034fea391281ef4ca68707c13</code>
<code>OwnbitMultiSigV11(Implementation).sol</code>	Accident Protection implementation	333	<code>4f6635331e3a88ce621f5425335ccbd24e6146f4e7dd2b0d0619bce0c3472a1e</code>
<code>OwnbitMultiSigV11(Proxy).sol</code>	Proxy for Standard and Accident Protection	26	<code>7cab4a2821047ce428b3ffe292775d5205224e33298962485ec5f32eb873ac81</code>
<code>OwnbitMultiSigV21(Implementation).sol</code>	Enterprise (weighted signers) implementation	448	<code>f017d4c4c82eea48567a876a741ee37b639dfb67a40a206cd616a0b1435d6b57</code>
<code>OwnbitMultiSigV21(Proxy).sol</code>	Enterprise proxy	53	<code>1a27bf59e0996dd907e8b69d2a7a659b0f5c68fee6d69827c9daf14e31deb878</code>
<code>ownbit-solana-multisig/src/lib.rs</code>	Solana multisig program (Anchor)	280	<code>8cd58c34d47316683cd45289d199907d939f95648d50268660789461561c59ef</code>

Deployment mapping (from the repository README and proxy constants)

Wallet type	Proxy source	Implementation source	Implementation address
Standard	V11 Proxy	V10	0x8ff6eBD112577E471530D175Ce151F1DcBDC4B8a
Accident Protection	V11 Proxy (alternate constant)	V11	0xAd63df1c78AaddC3Ee033308671Ed59ee19E20a8
Enterprise	V21 Proxy	V21	0x7f5c363b60C5304360E8956DE5B1aCe1EE8e94F9

Out of scope

- Earlier contract versions (V4 to V8, V20, `old/`), except as a baseline for differential review.
- The Ownbit mobile apps, the server-side signature coordination protocol, and the client-side wrapping of WalletConnect signatures.
- On-chain state: deployed bytecode matching, implementation initialization and the Solana upgrade authority. The review environment had no RPC access, so these are listed as verification items.
- The Solana build manifest (`Cargo.toml` , `Anchor.toml` , `Cargo.lock`), which is not in the repository.
- Formal verification and economic analysis beyond OMS-02.

Methodology and Severity Classification

Each finding is rated by impact and likelihood. Five EVM findings (OMS-01, OMS-02, OMS-04, OMS-08, OMS-09) were confirmed with executable tests; the Solana findings rest on manual analysis.

1. **Manual review.** Every in-scope line was read against known multisig failure classes: signature validation, replay, initialization, delegatecall proxies, external-call handling, threshold logic, Solana account validation and CPI.
2. **Differential review.** V21 was diffed against V20 to isolate the September 2026 changes: `block.chainid`, the non-reverting `spend()` and the ERC-1271 block.
3. **Proof-of-concept tests.** Foundry 1.5.1 with solc 0.8.28 (via-IR, optimizer 200 runs, EVM Cancun) ran 9 tests, all passing. Proxies were exercised by placing implementation bytecode at the constant addresses. An Anvil local node was used to measure

`eth_estimateGas` behaviour for OMS-02.

4. **Solana.** The program was reviewed manually. It was not compiled, because the repository has no build manifest.

Severity matrix

Likelihood	High impact	Medium impact	Low impact
High	Critical	Major	Minor
Medium	Major	Medium	Minor
Low	Medium	Minor	Informational

Severity definitions

Severity	Meaning
Critical	Direct and likely loss of funds or control; fix before further use.
Major	Loss of funds or control under specific conditions, a silent failure of core function, or a centralization risk; fix promptly.
Medium	Weakens a safety assumption or affects core function; direct loss unlikely without other failures.
Minor	Low impact: compatibility, defence in depth, verifiability.
Informational	Code quality, documentation or best practice; no direct security impact.

Status definitions: Pending (not yet addressed), Acknowledged (accepted without a code change), Partially Resolved, Resolved.

Findings Overview

Sixteen findings, all Pending at report date. Status is a dropdown so remediation can be tracked here.

ID	Title	Component	Severity	Status
OMS-01	Dormant wallet can be drained by any single owner after 416 days	V11	Major	Pending
OMS-02	Non-reverting <code>spend()</code> lets gas estimation or a third party burn the nonce on heavy calls	V10, V11, V21	Medium	Pending
OMS-03	Approvals do not commit to transaction content	Solana	Medium	Pending
OMS-04	Owner quorum can replace signers and spend with no delay (centralization)	V21	Major	Pending
OMS-05	Program upgrade authority and build reproducibility unverified (centralization)	Solana	Major	Pending
OMS-06	Approved transactions never expire and cannot be cancelled	Solana	Medium	Pending
OMS-07	Spend votes are <code>personal_sign</code> signatures over an opaque hash	V10, V11, V21	Medium	Pending
OMS-08	Proxy cannot receive ETH through a 2300-gas stipend when the implementation is cold	V11 Proxy, V21 Proxy	Minor	Pending
OMS-09	Implementation contracts can be initialized by anyone	V10, V11, V21	Minor	Pending
OMS-10	Deployed bytecode is hard to reproduce from source	V10, V11, V21, proxies	Minor	Pending
OMS-11	Transaction accounts are never closed and no events are emitted	Solana	Minor	Pending
OMS-12	Spend events omit the nonce and are inconsistently indexed	V10, V11, V21	Informational	Pending

ID	Title	Component	Severity	Status
OMS-13	Inconsistent signature checks and documentation errors	V10, V11, V21	Informational	Pending
OMS-14	Execution flag is written after the CPI	Solana	Informational	Pending
OMS-15	Unbounded return-data copy in <code>spend()</code>	V10, V11, V21	Informational	Pending
OMS-16	Encoding and gas nits	V11, V21	Informational	Pending

Detailed Findings: EVM Contracts

Five EVM findings were reproduced with executable tests; test names refer to the PoC suite listed in the appendix.

OMS-01 Dormant wallet can be drained by any single owner after 416 days

Category	Severity	Location	Evidence
Logic / Access control	Major	<code>OwnbitMultiSigV11(Implementation).sol</code> L139-179, L299-332	<code>DormancyTest</code> (3 tests)

Description. The Accident Protection rule lowers the spending threshold to the number of owners active in the last 416 days, with a minimum of 1 (`getRequiredWithoutInactive()`). Activity is refreshed only by an on-chain `spend()` or `active()` call.

The re-check after `_updateActiveTime()` stops an inactive owner from out-voting active ones. It does not help when every other owner is dormant: a 2-of-3 wallet with no transaction for 416 days becomes 1-of-3, and any one owner key can move all funds at once, with no delay and no chance for the others to object.

Long-term cold storage is exactly the wallet that sits untouched for over a year, so the rule weakens the wallets that most need multisig protection. The ERC-1271 path applies the same lowered threshold, so a single owner can also produce a valid off-chain signature, such as a Permit2 approval, that leaves no on-chain trace until used.

Proof of concept.

- `test_OMS01_singleKeyBeforeDormancyReverts` : one signature from owner 3 reverts.
- `test_OMS01_singleKeyAfterDormancyDrains` : after 416 days and 1 second, the same single signature moves 100 ETH.
- `test_OMS01_single1271SignatureBecomesValid` : a single-owner ERC-1271 signature returns `0xffffffff` before dormancy and `0x1626ba7e` after.

Attack scenario. An attacker finds one seed backup of a dormant 2-of-3 vault, waits until no owner has transacted for 416 days, then signs alone. An attacker holding a key early can call `active()` now and then to keep it live while the honest owners go quiet.

Recommendation.

- Client, no contract change: warn every owner well before day 416, offer a one-tap `active()` transaction, and show the current effective threshold on the wallet screen.
- Next version: route degraded-threshold spends through propose, wait (for example 30 days), then execute, cancellable by any owner signature or `active()` call.
- Next version: keep the full threshold in `isValidSignature()` ; off-chain approvals should never benefit from the accident rule.
- Consider a designated recovery key behind a time lock, as in miniscript decaying multisig, instead of "any owner".

OMS-02 Non-reverting `spend()` lets gas estimation or a third party burn the nonce on heavy calls

Category	Severity	Location	Evidence
Logic / Denial of service	Medium	V10 L107–118; V11 L119–130; V21 L163–187	<code>GasGriefingTest</code> (2 tests), Anvil experiment

Description. Since 2026-09-07, `spend()` increments `spendNonce` and emits `SpendFailed` instead of reverting when the inner call fails. The signed message carries no gas amount, and the call forwards all but 1/64 of the remaining gas (EIP-150). With too little gas, the inner call runs out while the retained 1/64 still pays for `SpendFailed` : the transaction succeeds, the nonce is consumed, and the intended call never ran. V20 reverted on failure and was not affected.

Two routes reach this state:

- **Gas estimation.** `eth_estimateGas` searches for the lowest gas limit that does not revert. Because the failure path no longer reverts, the search can settle on a limit at which the inner call fails. Anvil did so in the experiment below; Geth narrows its search around the gas actually used, which reduces but does not remove the risk.
- **Third-party resubmission.** `spend()` has no caller restriction, so anyone who sees the signed calldata in the mempool can resubmit it with a tuned gas limit.

In follow-up tests, only inner calls needing at least about 126,000 gas (V10, V11) or about 206,000 gas (V21) were exposed. The V21 figure rises by about 500 gas per byte of call data, because `SpendFailed` logs the data. ETH and token transfers are not exposed. That is precisely the class `WalletConnect` brings in: swaps, bridges, staking and lending.

Proof of concept.

- `test_OMS02_heavyCallCanBeGriefed` : a 2-of-3 V10 wallet signs a call needing about 360,000 gas. A third party resubmits it with 175,000 execution gas; it succeeds, emits `SpendFailed`, advances the nonce, and the owners' signatures become unusable.
- `test_OMS02_lightCallNotGriefable` : across 20,000 to 400,000 gas, a light call never fails while the outer call succeeds.
- Anvil experiment: `eth_estimateGas` returned 203,945 for the heavy spend. Sent with that limit and with common safety multipliers, every transaction succeeded on-chain while the target call failed.

Gas limit	Multiplier	Tx status	Nonce after	Target call ran
203,945	1.0x	Success	1	No
244,734	1.2x	Success	1	No
305,917	1.5x	Success	1	No
346,706	1.7x	Success	1	No
407,890	2.0x	Success	1	No

Impact. No funds are stolen, but the failure is invisible at the transaction level. Owners must re-sign, and transactions pre-signed for later nonces execute against a state their signers did not expect, for example a step that depended on an earlier DeFi action. Rated Medium: no funds are lost, and exposure depends on the node and on the app's gas strategy.

Recommendation.

- Now, in the client: never use `eth_estimateGas` on `spend()` directly. Estimate the inner call from the wallet address to the destination, set the outer limit to at least 64/63 of that plus the multisig overhead, simulate the final transaction, and refuse to broadcast if the simulation emits `SpendFailed`.
- Show `SpendFailed` to users as a failed transaction, and submit owner transactions through a private RPC where available.
- Next version: include a signed `minGas` (Safe's `safeTxGas`) and enforce `require(gasleft() >= max(minGas * 64 / 63, minGas + 2500) + 500)` before the call, as Safe does.

OMS-04 Owner quorum can replace signers and spend with no delay (centralization)

Category	Severity	Location	Evidence
Centralization / Privilege	Major	V21 L274–313, L163–187	<code>EnterpriseTakeoverTest</code>

Description. In the Enterprise wallet, `changeSigners()` needs only `ownerRequired` owner signatures. The new signer set takes effect in the same transaction, and spending then needs only the new signers. The real protection of the funds is therefore the weaker of the owner quorum and the signer quorum, not the signer quorum users see. This behaviour is documented and already existed in V20, and `changeOwners()` follows the same rule.

Proof of concept. `test_OMS04_adminReplacesSignersThenSpendsAlone`: a wallet with a 1-of-1 admin and 3-of-5 signers. The admin makes itself the only signer and moves 100 ETH, in two calls in the same block.

Recommendation.

- Show the effective security level, the weaker of the two quorums, at wallet creation and on the wallet screen, with a warning when the owner quorum is the weaker one.
- Next version: time-lock configuration changes (for example 48 hours to 7 days) with a signer-quorum veto, and emit a pending-change event for monitoring.
- Alternatively require signer approval for signer-set changes and keep owners as a delayed recovery path.

OMS-07 Spend votes are `personal_sign` signatures over an opaque hash

Category	Severity	Location	Evidence
Signature scheme	Medium	V10 L85–96; V11 L97–108; V21 L131–150	Manual analysis

Description. A spend vote is an EIP-191 `personal_sign` signature over a 32-byte keccak hash of packed fields. This has two consequences.

- Signing devices can only display an opaque hash, so owners sign blind unless the client recomputes and shows the fields.
- Any owner (V10, V11) or signer (V21) key that ever `personal_sign`s an attacker-chosen 32-byte message, for example in a dApp login or in another wallet after a key export, produces a valid spend vote for whatever destination and value the attacker hashed.

The ERC-1271 wrapping protects the dApp-signing path that Ownbit controls. It cannot protect an owner key used anywhere else, and the contracts leave that duty to the client. The source comments at V10 L172–174, V11 L227–229 and V21 L363–365 say a `personal_sign` signature cannot become a spend authorisation; that holds only for messages the Ownbit client wraps. In V21 the same reuse applies to owner keys, whose votes authorise configuration changes.

Recommendation.

- Next version: sign spends as EIP-712 typed data, for example `SpendTransaction(address to,uint256 value,bytes data,uint256 nonce,uint256 minGas)` under the wallet's domain, so hardware and cold signers can show every field.
- Client: keep owner keys exclusive to multisig use, refuse `personal_sign` of exactly 32 bytes from an owner key, and have the cold signer recompute the hash from decoded fields before signing.

OMS-08 Proxy cannot receive ETH through a 2300-gas stipend when the implementation is cold

Category	Severity	Location	Evidence
Compatibility	Minor	V11 Proxy L15–24; V21 Proxy L34–43	<code>StipendTest</code> (2 tests)

Description. The proxies have no `receive()`, so plain ETH goes through `fallback()` and a delegatecall to the implementation. When the implementation address is cold in the transaction, that access alone costs 2,600 gas (EIP-2929), above the 2,300-gas stipend of Solidity `transfer()` and `send()`. Senders that still use them, such as some exchange withdrawal contracts and older payment contracts, revert. If the implementation is already warm, the transfer succeeds, which is why unwrapping WETH from the wallet itself works.

Recommendation. Next proxy version: add `receive() external payable { emit Funded(msg.sender, msg.value); }` in the proxy itself, with no delegatecall. For deployed wallets, document the limitation; a sender can also attach an EIP-2930 access list that includes the implementation.

OMS-09 Implementation contracts can be initialized by anyone

Category	Severity	Location	Evidence
Initialization	Minor	V10 L51-62; V11 L58-69; V21 L73-106	UninitializedImplTest

Description. `initialize()` is public and guarded only by an empty owner list. On the implementation contract itself storage is empty, so anyone can appoint themselves its owner. Proxies are not affected: they initialize in their constructor, and the implementations contain no `delegatecall` or `selfdestruct`. The exposure is limited to assets sent to an implementation address by mistake and to confusing explorer data.

Recommendation. Check whether the three deployed implementations are initialized and, if not, initialize them to an unusable configuration. Next version: lock the implementation in its constructor, for example by setting the threshold to `type(uint256).max`.

OMS-10 Deployed bytecode is hard to reproduce from source

Category	Severity	Location	Evidence
Verifiability	Minor	All EVM files	Build attempt

Description. Published source only builds trust if anyone can rebuild the deployed bytecode from it. Three things prevent that today:

- The V21 implementation does not compile with the legacy pipeline ("stack too deep" at L304). It needs via-IR, which the repository does not document; V20 fails the same way. V21 also uses `bytes.concat`, which needs Solidity 0.8.4 or later, above the declared floor of 0.8.0.
- Every file uses the floating pragma `>=0.8.0 <0.9.0`; the compiler version, optimizer runs and EVM version are not recorded.
- One proxy source serves two wallet types by commenting out an `implementation` constant (V11 Proxy L5–6), so the committed file matches only Standard wallets.

Recommendation. Pin the compiler, commit the build configuration or standard-JSON inputs, split the proxy into one file per wallet type, and verify every implementation plus a sample proxy per chain on Sourcify (full match) and the chain explorers. Link those verifications from the Ownbit security page.

OMS-12 Spend events omit the nonce and are inconsistently indexed

Category	Severity	Location	Evidence
Monitoring	Informational	V10 L35–41; V11 L42–48; V21 L38–42	Manual analysis

Description. Clients must now judge success from `Spent` and `SpendFailed`, yet neither event carries the nonce. V10 and V11 index no fields and omit the call data, so a client tracking several queued nonces cannot match an event to a nonce without decoding transaction input. V21's `OwnersChanged` and `SignersChanged` events also omit the config nonce.

Recommendation. Next version: `Spent(uint256 indexed nonce, address indexed to, uint256 value, bytes32 dataHash)`, and the same fields in `SpendFailed`.

OMS-13 Inconsistent signature checks and documentation errors

Category	Severity	Location	Evidence
Code quality	Informational	V10, V11, V21	Manual analysis

- `spend()` accepts high-s signatures and only `v` of 0 or 1; `isValidSignature()` rejects high-s and accepts `v` of 0, 1, 27 or 28. Malleability is harmless here because nonce and distinct-signer checks stop reuse, but two rules for one key set invite client bugs.

- In `spend()`, a `v` of 229 or more overflows `vs[i]+27` and reverts with a panic instead of a clear error.
- The V10 and V11 headers say "The S coordinate must be 0x00 or 0x01"; this describes V.
- The V11 comment on `Spent` refers to `spendAny`, which does not exist, and the README's "Known issue" heading is empty.

Recommendation. Use one validation helper for both paths (low-s, normalized `v`) and correct the comments.

OMS-15 Unbounded return-data copy in `spend()`

Category	Severity	Location	Evidence
Gas / Robustness	Informational	V10 L112; V11 L124; V21 L181	Manual analysis

Description. `(bool sent, bytes memory ret) = destination.call{value: value}(data)` copies all return data into memory and logs it in `SpendFailed`. A destination that reverts with a very large payload drives up memory and log costs, so the transaction becomes very expensive or reverts outright without consuming the nonce. This matters only when owners sign a call to a hostile contract.

Recommendation. Copy at most a fixed prefix of return data, for example 256 bytes, with inline assembly.

OMS-16 Encoding and gas nits

Category	Severity	Location	Evidence
Code quality	Informational	V11, V21, V21 Proxy	Manual analysis

- V21 `_signersChangeMessage` (L229–251) packs two dynamic arrays back to back. It is unambiguous only because `validSignerRequirement` forces equal lengths; `abi.encode` or EIP-712 removes the dependency.
- V21 `_verifyOwnerSigsCount` recomputes `_toEthSigned(message)` on every loop iteration (L264).
- V11 `active()` is `payable`; ETH sent through it emits no `Funded` event.
- V10 and V11 receiver hooks are not `pure` and have unused parameters, producing compiler warnings.

- V21 Proxy `_getRevertMsg` falls back to a generic message when revert data is under 68 bytes (panics, bare reverts, custom errors without arguments), and would misread a longer custom error as a string.
- `spend()` with non-empty data to an address that has no code succeeds and emits `Spent`, although nothing ran, so a mistyped or undeployed token address looks like a completed transfer. This was also true in V20; check `destination.code.length > 0` when data is non-empty.

Detailed Findings: Solana Program

The Solana program follows the Serum multisig design with owner-change instructions removed. Its account validation is careful; the risks lie in what an approval commits to and in who controls the program itself.

OMS-03 Approvals do not commit to transaction content

Category	Severity	Location	Evidence
Signature scheme / Trust	Medium	<code>lib.rs</code> L100–112 (<code>approve</code>), L192–199 (<code>Approve</code>)	Manual analysis

Description. The proposer's `create_transaction` carries the full instruction (program, accounts, data) as arguments, so the proposer's signer can see what it proposes. Every other owner calls `approve()`, which takes no arguments and names only the multisig, the transaction account and the owner. Because a transaction's content cannot change after creation, the approval does pin that content on-chain.

The gap is off-chain. An offline signer cannot read chain state, so it cannot check what the account it approves holds and must trust the online device's description. A compromised watch wallet, coordination server or proposer can show owners one transaction and collect approvals for another. On EVM each signer signs a hash of the actual destination, value and data, which a cold device recomputes itself. Rated Medium: exploitation requires compromising the online side first.

Recommendation. Next program version: `approve(expected_hash: [u8; 32])`, where the hash covers the multisig, program ID, accounts and data, and the instruction requires a match. The cold signer decodes the instruction from the QR payload, recomputes the hash, shows the fields, then signs. The same check can be added to `execute_transaction`.

OMS-05 Program upgrade authority and build reproducibility unverified (centralization)

Category	Severity	Location	Evidence
Centralization	Major (Critical if a single key)	Program 7uPTVnLrNM9bAdCGrfckz1ugJeRs5bEGcEgcwzPuFsAK ; repository	Not verifiable from the review environment

Description. Solana programs are upgradeable by default. Whoever holds the upgrade authority can replace the code that signs for every multisig PDA, and with it move the funds of every Ownbit Solana wallet without any owner signature. The review environment could not query the chain, so the authority's status is unknown.

The repository also lacks `Cargo.toml`, `Anchor.toml` and `Cargo.lock`. No one can rebuild the program and compare it with the deployed binary, so the published source cannot be tied to what runs on-chain. If the authority is a single key, Ownbit's Solana multisig is custodial in practice and this finding is Critical.

Recommendation.

- Check the authority with `solana program show` and publish the result.
- Either make the program immutable with `solana program set-upgrade-authority -final`, or move the authority to a multisig behind a public time lock.
- Commit the full Anchor workspace and lockfile, and publish a verifiable build, for example with the `solana-verify` CLI.

OMS-06 Approved transactions never expire and cannot be cancelled

Category	Severity	Location	Evidence
Logic	Medium	<code>lib.rs</code> L115-174; no cancel or close instruction	Manual analysis

Description. A transaction that reaches the threshold stays executable forever, by anyone. There is no expiry, no cancel instruction and no sequence number, and executing a newer transaction leaves older ones untouched. The EVM contracts avoid this because each nonce is usable once, in order.

Scenario. Owners approve a 100 SOL payment that fails because the vault holds only 50 SOL. They approve and execute a replacement for 50 SOL. Months later the vault is refilled, and the recipient executes the old transaction to be paid a second time. Likewise,

a proposal carrying a stolen key's vote can wait indefinitely for one more approval obtained by social engineering.

Recommendation. Add a stale-transaction index on the multisig account, checked in `approve` and `execute_transaction` and raised by a threshold-approved "invalidate pending" action (Squads v4 uses a similar index for configuration changes). Add a threshold-approved cancel, and consider an optional expiry slot.

OMS-11 Transaction accounts are never closed and no events are emitted

Category	Severity	Location	Evidence
Resources / Monitoring	Minor	<code>lib.rs</code> (whole program); <code>ErrorCode</code> L21-25	Manual analysis

Description. Each proposal is a new account whose rent deposit, paid by the proposer, is never returned; executed transactions stay on-chain with their lamports. The error codes `TransactionAlreadySigned`, `UnableToDelete` and `Overflow` are declared but never used, which suggests a delete path was planned and dropped. The program emits no events, so monitoring must parse instruction data.

Recommendation. Add a `close` for executed or cancelled transactions that refunds the proposer, emit Anchor events on create, approve and execute, and remove unused error codes.

OMS-14 Execution flag is written after the CPI

Category	Severity	Location	Evidence
Code quality	Informational	<code>lib.rs</code> L168-171	Manual analysis

Description. `did_execute` is set after `invoke_signed` and persisted only when the instruction exits, so code reached through a self-CPI during the inner call still sees `false`. A double execution would need the stored instruction to execute its own transaction, which recurses until the CPI depth limit and fails; no exploit was found. Checks-effects-interactions remains the safer order.

Recommendation. Set `did_execute = true` and persist it before the CPI (for example with the account's `exit()`), or reject transactions whose `program_id` is this program.

Centralization and Trust Assumptions

The EVM wallets have no admin key and no upgrade path. The remaining trust sits in the Ownbit client, the signature-coordination server and, on Solana, whoever holds the program's upgrade authority.

Assumption	Trusted party	If violated	Related
EVM implementation code cannot change	None: the address is a constant in proxy bytecode	Not applicable; this is a strength	—
Solana program code cannot change maliciously	Upgrade-authority holder	Every Solana vault can be drained	OMS-05
Client wraps dApp messages with <code>getMessageHash</code> and rejects <code>eth_sign</code>	Ownbit app	A dApp could obtain a spend vote	OMS-07
Signers see the real content of what they sign	App, watch wallet and server	Owners approve a different transaction	OMS-03
Client sets gas so heavy calls do not silently fail	Ownbit app	Nonce burned, intended call skipped	OMS-02
Enterprise owners act only as admins	Owner quorum	Owners replace signers and spend	OMS-04
Accident Protection owners transact every 416 days	Each owner	One key controls the vault	OMS-01

Design observations

- **Escape path without Ownbit.** `spend()` and the Solana instructions accept valid signatures from any submitter. If Ownbit's server disappeared, owners could still move funds with any tool that builds the same messages. Publishing a standalone recovery tool and the message formats would turn this property into a user-visible guarantee.
- **ERC-1271 bypasses the nonce queue.** Off-chain signatures such as Permit2 approvals or marketplace orders can move assets without passing through `spend()`, and they cannot be revoked by the wallet except, in V21, by changing signers. The client must decode typed data and flag token-approval permits before signing.

- **Fixed owner sets.** V10, V11 and the Solana program cannot rotate keys. A compromised key stays an owner for the life of the wallet, and losing keys beyond the threshold locks funds permanently; V11's remedy for this is the dormancy rule behind OMS-01. The supported mitigation is migrating funds to a new wallet, which the app should make a first-class, guided flow.

Positive Observations and Recommendations

Six of the eleven recommended actions need no contract change and can ship in the app now; the rest belong in the next contract and program versions.

Positive observations

- **Replay protection.** Every EVM digest binds the wallet address, a per-wallet nonce and the live `block.chainid`, read at execution so it stays correct after a chain split.
- **Domain separation.** Spend votes use the EIP-191 prefix, ERC-1271 uses an EIP-712 domain bound to chain and wallet, and V21 tags owner and signer changes with distinct domain values and a separate config nonce.
- **Signature hygiene.** Recovered addresses must be distinct, current owners or signers; zero addresses are rejected at initialization and after `ecrecover`.
- **Checks-effects-interactions.** `spend()` increments the nonce before the external call, so re-entry cannot reuse signatures.
- **Minimal immutable proxies.** The implementation address is a constant, its code is checked at deployment, and initialization happens once, in the proxy constructor.
- **Solana account validation.** The canonical bump is enforced; multisig and transaction accounts must sign their own creation, which prevents initialization front-running; executed accounts must exactly match the stored list; `has_one` binds each transaction to its multisig.

Recommended actions

When	Action	Findings
Now (app)	Estimate the inner call, simulate the final transaction, treat <code>SpendFailed</code> as failure	OMS-02
Now (app)	Remind owners before day 416, offer one-tap <code>active()</code> , show the effective threshold	OMS-01

When	Action	Findings
Now (app)	Show Enterprise users the weaker of the owner and signer quorums	OMS-04
Now (app)	Refuse 32-byte <code>personal_sign</code> from owner keys; decode and flag permit typed data	OMS-07
Now (ops)	Check and publish the Solana upgrade authority; make it immutable or time-locked	OMS-05
Now (ops)	Initialize the deployed implementations; pin builds and verify on Sourcify and explorers	OMS-09, OMS-10
Next EVM version	EIP-712 spend message with a signed <code>minGas</code> and a <code>gasleft()</code> check	OMS-02, OMS-07
Next EVM version	Challenge window for degraded-threshold spends; full threshold for ERC-1271	OMS-01
Next EVM version	Time-locked configuration changes with signer veto	OMS-04
Next EVM version	Proxy <code>receive()</code> , locked implementation, nonce in events, capped return data	OMS-08, OMS-09, OMS-12, OMS-15
Next Solana version	Content-hash <code>approve</code> , stale index, cancel and close, events, flag set before CPI	OMS-03, OMS-06, OMS-11, OMS-14

Before deploying the next versions, commission an independent third-party audit and publish it with the verified builds.

Appendix

A. Proof-of-concept tests. All 9 tests pass on Foundry 1.5.1 with solc 0.8.28 (via-IR, optimizer 200 runs, EVM Cancun). The test file `OwnbitPoC.t.sol` accompanies this report; place the five in-scope Solidity files in `src/` as `V10Impl.sol`, `V11Impl.sol`,

`V11Proxy.sol`, `V21Impl.sol` and `V21Proxy.sol`, then run `forge test -vv`.

Test	Finding	Result
<code>test_OMS01_singleKeyBeforeDormancyReverts</code>	OMS-01	Pass
<code>test_OMS01_singleKeyAfterDormancyDrains</code>	OMS-01	Pass
<code>test_OMS01_single1271SignatureBecomesValid</code>	OMS-01	Pass
<code>test_OMS02_heavyCallCanBeGriefed</code>	OMS-02	Pass
<code>test_OMS02_lightCallNotGriefable</code>	OMS-02	Pass
<code>test_OMS04_adminReplacesSignersThenSpendsAlone</code>	OMS-04	Pass
<code>test_OMS08_transferFailsWhenImplementationCold</code>	OMS-08	Pass
<code>test_OMS08_transferWorksWhenImplementationWarm</code>	OMS-08	Pass
<code>test_OMS09_anyoneCanInitializeImplementation</code>	OMS-09	Pass

B. On-chain verification checklist. These checks were out of reach of the review environment and should be completed by the Ownbit team.

- ☐ Record the upgrade authority of `7uPTVnLrNM9bAdCGrfckz1ugJeRs5bEGcEgcwzPuFsAK` with `solana program show` (OMS-05).
- ☐ Call `getOwners()` on each implementation address on Ethereum, BNB Smart Chain, Base and Arbitrum One to confirm whether it is initialized (OMS-09).
- ☐ Rebuild each implementation with pinned settings and compare it with the deployed bytecode (OMS-10).
- ☐ Confirm how the app sets the gas limit for `spend()` (OMS-02).

C. Disclaimer. This report covers only the files and commit listed under Audit Scope. Code changed or deployed after 2026-10-26 is not covered. It was prepared by an AI reviewer (GPT-6-Astra) and has not been reviewed by an accredited third-party security firm. It does not guarantee the absence of vulnerabilities, does not endorse the project, and is not investment advice.