CMTAT v3.0.0-rc5 - Audit Agent Report - Comment

This tab summarizes the finding the reported by the Nethermind Audit Agent on CMTAT codebase, version v3.0.0-rc5.

N°, Title and Severity columns come from the agent, while the others are the answers by CMTA maintainers.

N°	Title	Severity (Agent)	Validity [Valid, Invalid, Partial]	Design choice	Status [Fixed/Pending/Documented/-]	Comment
1	CMTATUpgradeableUUPS contract is not initializable	Hight	Partial	☒	Fixed in the next version (v3.1.0) See issues/327 & Commit	Only a concern if the proxy is not initialized at deployment, which is not recommended because their is a risk of front-running if the proxy is initialized later.
2	batchTransfer bypasses all compliance / enforcement logic	Hight	Valid	☒	Fixed (v.3.0.0)	Fixed before the final version CMTAT v3.0.0. Severity should be medium because the function is restricted to the MINTER_ROLE
3	Upgrade authorisation can be performed by Default Admin even without PROXY_UPGRADE_ROLE	Medium	Valid	☑	Documented	The admin has all the right on the contract, including upgrading the proxy. In the documentation, we decided to hihglight that and suggestion to potential CMTAT use to separate the two roles
4	Trusted-forwarder address not validated – a malicious forwarder can impersonate any user	Medium	Invalid	☑	Documented	The forwarder is irrevocable by design choice from OpenZeppelin to avoid notably the setting of a malicious forwarder after deployment Contract deployer must check that the forwarder address supplied is a trusted one.
5	Missing Initialization for AllowlistModule	Medium	Invalid	☑	-	This is intended. Also, if someone uses CMTAT allowlist version, we estimate that this is notably to restrict the transfers.
6	Denial of Service via Unbounded Loop in Snapshotting	Low	Valid	☑	-	This concerns a mock contract. But in any case, an address with the SNAPSHOOTER_ROLE is supposed to be trusted.

N°	Title	Severity (Agent)	Validity [Valid, Invalid, Partial]	Design choice	Status [Fixed/Pending/Documented/-]	Comment
7	Immutable trusted forwarder creates upgrade challenges	Low	Invalid	☑	-	See finding 4
8	Missing validation for zero decimals requirement	Low	Invalid	☑	-	We let CMTAT users to configure the decimals of their choice for flexibility
9	Missing Input Validation in batchSetAddressAllowlist Function	Low	Valid	☑	-	Checking address increases gas cost unnecessary
10	Missing Input Validation in setAddressAllowlist Function	Low	Valid	☑	-	See finding 9
11	Incorrect error parameters in _unfreezeToken	Info	Valid	☒	Fixed in the next version (v3.1.0) See issues/329 & Commit	-
12	Uninitialized variable used in condition	Info	Valid	☒	Pending issues/3	It concerns a mock contract now available in another repository.
13	Potential Front-Running in Allow/Blocklisting	Info	Valid	☑	-	No real solution available, concerns also Tether, Circle and other big players in the space.
14	Potential for External Engine Integration Failures	info	Valid	☑	-	Verifying the functionality of the engines must be done before setting them in the contract
15	Lack of Input Validation in freezePartialTokens and unfreezePartialTokens	Best practice	Valid	☑	-	-
16	Missing Zero Address Validation for Trusted Forwarder	Best practice	Invalid	☑	-	See finding 4
17	Insufficient Input Validation for Debt Module Functions	Best practice	Valid	☑	-	No input validation because otherwise we reach the maximum contract size limit on Ethereum
18	Inconsistent Pause Protection	Best practice	Valid	☑	Documented	burn and mint can be done even if the contract is in the pause state, but for crosschainburn and crosschainmint, we decided that it is safer to apply the pause modifier
19	Misleading NatSpec Comments in Deployment Contracts	Best practice	Valid	☒	Fixed in the next version (v3.1.0) CMTAT/issues/330 & Commit	-
20	Missing Granular Events for Important State Changes	Best practice	Invalid	☑	-	We have other custom events in top level functions
21	Misleading NatSpec comments in constructors of upgradeable contracts	Best practice	Valid	☑	See finding 19	-
22	Use of block.timestamp for Time-Sensitive Operations	Best practice	Invalid	☑	-	-