CMTAT - Audit-20210910

Introduction

This document concerned the audit report ABDK-CMTAT-audit-20210910.pdf which was made on the project CMTAT.

The audited version is that of commit cd3af7

Vocabulary

Header	Description
Id	Id of the concerned CVF
Severity	Same notation that in the report Color : Minor / Moderate / Major
Object	The concerned object [Functions, variables, event name, etc..]
Decision	Which decision has been taken and why
Status	[Fixed] => CVF has been fixed [In progress] => CVF is being fixed (pull request / issue) [Open] => CVF is NOT fixed [Closed] => CVF is not more relevant or a decision to not fix the CVF has been taken Fixed / In progress / Open / Closed
Commit / pull request fix	The concerned commit / pull request on github which fix the CVF
Remark	Any other relevant information, by example suggestions to fix the CVF.

Summary

ID	Severity	Module	Object [Functions, variables, event name, etc..]	Description (evt recommendation)	Decision	Status	Commit / pull request fix	Remark
1	Minor	SnapshotModule.sol All modules are concerned	pragma solidity	Solidity version		Opened		See issues/68
2	Minor	SnapshotModule.sol	-	The version of the imported library should be provided.		Opened		We can indicate the version in the file TOOLCHAIN
3	Minor	SnapshotModule.sol	-	Provide a short function description and parameter meaning in the comments preceding the function.		Opened		
4	Minor	SnapshotModule.sol	SNAPSHOTER_ROLE	The word “snapshoter” (12K results in Google) sounds odd.	The variable is renamed in snapshooter	Fixed	pull/13 Commit	
5	Minor	SnapshotModule.sol	_currentSnapshot	This variable is not initialized.	explicitly initializing to 0 Fixed	Fixed	pull/14 Commit	The commit fix broke the proxy construction. Commit fix
6	Moderate	SnapshotModule.sol	_scheduledSnapshots	Using an unordered list of scheduled snapshots and removing already created snapshots from it is suboptimal		Opened		

				and have several important drawbacks [...]				
7	Minor	SnapshotModule.sol	scheduleSnapshot	Function scheduleSnapshot always returns its argument.	No return value	Opened	Pull/76 Commit	
8	Minor	SnapshotModule.sol	_rescheduleSnapshot	Here two passes over the array are made while just one suffices.		Opened		
9	Minor	SnapshotModule.sol	_rescheduleSnapshot	This function always returns its second argument which is clearly redundant.	No return value	Fixed	Pull/76 Commit	
10	Minor	SnapshotModule.sol	_unschedule Sna pshot	function always returns its argument which is clearly redundant	No return value	Fixed	Pull/76 Commit	
11	Minor	SnapshotModule.sol	getNextSnapshots()	This method returns also uncleared snapshots from the past.	-	Opened		
12	Minor	SnapshotModule.sol	SnapshotTotalSupply snapshotTotalSupply	Is it desired behaviour that the current balance is returned for an invalid snapshot time?	-	Opened		
13	Moderate	SnapshotModule.sol	_setCurrentSnapshot	This call reads the entire snapshot array, which is a significant overhead over each transfer.	-	Opened		
14	Minor	SnapshotModule.sol	_beforeTokenTransfer	The most often used branch is the last one.	refactoring order of conditions as indicated in the audit	Fixed	pull/16 Commit	

					report			
15	Minor	SnapshotModule.sol	_valueAt	Semantics of returned values is unclear		Opened		
16	Minor	SnapshotModule.sol	_getCurrentSnapshot	This function is redundant as it just returns the value of a storage variable.		Opened		
17	Minor	SnapshotModule.sol	Several functions (see report)	These loops do linear searches through the array whose length is unlimited. This makes the contract vulnerable to DoS attacks by scheduling a large number of snapshots.		Opened		
18	Minor	SnapshotModule.sol	_clearPastScheduled()	The "_scheduledSnapshots.length" value is read on every iteration.		Opened		
19	Minor	SnapshotModule.sol	_clearPastScheduled()	The element to be moved to this position will be once again read from storage at the next iteration		Opened		
20	Minor	SnapshotModule.sol	_removeScheduledItem	This line executes even if index is the last element.		Opened		
21	Minor	SnapshotModule.sol	event RuleEngineSet	The parameter type should be "IRuleEngine".		Opened		
22	Minor	ValidationModule.sol	IRule Engine pub lic rule Engine	This module should use "IRule" instead of "IRuleEngine" as it doesn't need		Opened		

				to know that the rule used is actually a composition of several nested rules.				
23	Minor	ValidationModule.sol	IRule Engine public ruleEngine	The module doesn't allow changing the rule engine after the deployment.		Opened		
24	Minor	ValidationModule.sol	_validateTransfer	This will revert in case the rule engine is not set.		Opened		
25	Minor	ValidationModule.sol	_messageForTransferRestriction	This will revert in case the rule engine is not set.		Opened		
26	Minor	ValidationModule.sol	_detectTransferRestrictionc	This will revert in case the rule engine is not set.		Opened		
27	Minor	CMTAT.sol	TRANSFER_OK	Defining restriction code constants in several placed is error-prone.	Define an enum in CMTAT.sol	Fixed	Pull/78 Commit	
28	Moderate	CMTAT.sol	__CMTAT_init	There should be a call to the "__Validation_init_unchained" function somewhere after this call.		Opened		
29	Minor	CMTAT.sol	__CMTAT_init	There should be a call to the "__ERC165_init_unchained" function before this call.		Opened		
30	Minor	CMTAT.sol	__CMTAT_init	This should be done before the "__Base_init_unchained" call as Base module inherits from ERC20		Opened		

				module.				
31	Minor	CMTAT.sol	__CMTAT_init	There should be a call to the “__ERC2771Context_init_unchained” function before this call.	No more relevant because __ERC2771Context_init_unchained” do not exist anymore in the recent version of OZ	Closed		
32	Minor	CMTAT.sol	mint	This event should be emitted inside the “_mint” function.		Opened		
33	Minor	CMTAT.sol	burnFrom	This would emit the “Approval” event (which is not desired) but will not emit the “Spend” event (which is actually desired).		Opened		
34	Minor	CMTAT.sol	burnFrom	This event should be emitted inside the “_burn” function.		Opened		
35	Minor	CMTAT.sol	Several functions (see report)	With multiple inheritance, it is unclear what base contract is referred as “super” here.		Opened		
36	Minor	CMTAT.sol	DetectTransferRestriction messageForTransferRestriction	This logic should be moved to the “PauseModule” contract.		Opened		

37	Minor	CMTAT.sol	detectTransferRestriction	Should be “} else” for readability.	Proposed changes have been implemented	Fixed	Pull/17 Commit	
38	Minor	CMTAT.sol	DetectTransferRestriction messageForTransferRestriction	This logic should be moved to the “EnforcementModule” contact.		Opened		
39	Minor	CMTAT.sol		This logic should be moved to the “ValidationModule”.		Opened		
40	Minor	CMTAT.sol	setTokenId setTerms ~~set TrustedForwarder~~	These functions should log some event.	Define events setTokenId & setTerms	Fixed	Pull/80 Commit	
41	Minor	CMTAT.sol	kill	Due to various misuse cases, it is best practice to use selfdestruct only when multiple short-lived contracts are created		Opened		
42	Minor	CMTAT.sol	_beforeTokenTransfer	These checks should be the first to save gas.	Proposed changes have been implemented	Fixed	Pull/18	
43	Minor	CMTAT.sol	_beforeTokenTransfer	This check should be done in the “PauseModule” smart contract.		Opened		
44	Minor	CMTAT.sol	_beforeTokenTransfer	This check should be done in the “EnforcementModule” smart		Opened		

				contract.				
45	Minor	CMTAT.sol	_beforeToken Transfer	This code should be moved to the “ValidationModule” smart contract.		Opened		
46	Minor	PauseMod ule.sol	-	The version of the imported library (OZ) should be provided.		Opened		We can indicate the version in the file TOOLCHAIN
47	Minor	PauseMod ule.sol	-	This contract should defines the “__Pause_init” and “__Pause_init_unchained” functions.		Opened		
48	Minor	PauseMod ule.sol	TRANSFER_ REJECTED_P AUSED	Defining constants for different transfer rejection reasons in different contracts is error-prone	Define an enum in CMTAT.sol	Fixed	Pull/78 Commit	
49	Minor	Authorizati onModule. sol	-	The version of the imported library (OZ) should be provided.		Opened		We can indicate the version in the file TOOLCHAIN
50	Minor	Authorizati onModule. sol	-	It is a good practice to put a comment into an empty block to explain why it is empty.		Opened		
51	Minor	Authorizati onModule. sol	-	This contract should define the “__Authorization_init” and “__Autho- rization_init_unchained” functions.		Opened		
52	Minor	MintModul e.sol	-	This module doesn’t actually implement minting functionality.		Opened		
53	Minor	Enforceme		The version of the imported		Opened		We can indicate the version

		ntModule.sol		library (OZ) should be provided.				in the file TOOLCHAIN
54	Minor	EnforcementModule.sol	F r e e z e U n f r e e z e	These events should probably also have the “reason” parameter.		Opened		
55	Minor	EnforcementModule.sol	TRANSFER_REJECTED_FROZEN	Defining constants for different transfer rejection reasons in different contracts is error-prone.	Define an enum in CMTAT.sol	Fixed	Pull/78 Commit	
56	Moderate	EnforcementModule.sol	TEXT_TRANSFER_REJECTED_FROZEN	The message is exactly the same as in the “PauseModule” contract. Also, the message is misleading.	The text message has been changed	Fixed	Pull/79 Commit	
57	Minor	MetaTxModule.sol	-	The version of the imported library (OZ) should be provided.		Opened		We can indicate the version in the file TOOLCHAIN
58	Minor	MetaTxModule.sol	__MetaTx_init	This code relies on how the base contract is implemented.	No more relevant because __ERC2771Context_init_unchained” do not exist anymore in the recent version of OZ	Closed		
59	Major	MetaTxModule.sol	__MetaTx_init_unchained	An unchained initializer is not supposed to call the base contract initializer.	The problematic function call was removed.	Fixed	Commit	

60	Minor	BurnModule.sol	-	The version of the imported library (OZ) should be provided.		Opened		We can indicate the version in the file TOOLCHAIN
61	Minor	BurnModule.sol	-	This module doesn't actually implement burning. Also, in inherits from the "Initializable" interface but doesn't have any initializer functions		Opened		
62	Minor	BaseModule.sol	-	The version of the imported library (OZ) should be provided.		Opened		We can indicate the version in the file TOOLCHAIN
63	Minor	BaseModule.sol	-	This requirement is not present in the module documentation	The requirement was added to the documentation in base.md	Fixed	Pull/19 Commit	
64	Major	BaseModule.sol	transferFrom	The returned value is ignored.	Proposed changes have been implemented	Fixed	Commit	
65	Major	BaseModule.sol	approve	It would be better to call "super.approve" here. The current code relies on the knowledge of how the base contract is implemented.	Proposed changes have been implemented	Fixed	Commit	
66	Minor	IRuleEngine.sol	-	It is common practice to provide a short function description and parameter meaning in the comments preceding the function.	Proposed changes have been implemented	Fixed	Commit	
67	Minor	IRuleEngin	-	This interface is redundant. It		Opened		not fixed, it is no real

		e.sol		basically defines a composite rule, i.e. a rule composed from other rules, but for those tokens that use such composite rule, it's composite nature doesn't not make any difference and is basically an implementation details.				necessary to fix this CVF, it is more clear to separate IRuleEngine and IRule
68	Minor	IRuleEngine.sol	setRules	Setting all rules at once effectively limits the maximum number of rules, as size of a transaction is limited by block gas limit.		Opened		not fixed, the remove functionality makes the code more complicated.
69	Minor	IRuleEngine.sol	ValidateTransfer detectTransferRestriction messageForTransferRestriction	These functions are very similar to the functions defined in the "IRule" interface.	Create two interfaces IERC1404 & IERC1404Wrapper (initially called IERC1404Common)	Fixed	Commit-1 Commit-2	
70	Minor	IRule.sol	-	It is common practice to provide a short function description and parameter meaning in the comments preceding the function.	Proposed changes have been implemented	Fixed	Commit	
71	Minor	IRule.sol	IRule	The interface name is too generic. The interface is all about transfers, consider reflecting this in the		Opened		

				name.				
72	Minor	IRule.sol	detectTransferRestriction canReturnTransferRestrictionCode messageForTransferRestriction	The semantics of these functions is unclear from their signatures.	Add a short description + create interface IERC1404	Fixed	Commit	
73	Minor	IRule.sol	DetectTransferRestriction canReturnTransferRestrictionCode messageForTransferRestriction	Types narrower than 256 bits don't save gas when used with function arguments or return values, however they limit the range of possible values.	No change The original goal was to be compliant with EIP-1404: ethereum/EIPs#1404 and restrictionCode is uint8	Closed	See pull/70	Remark on the save gas part The parameter argument is linked with variable stored in storage and in storage use uint8 rather than 256 can save gas