CERTIK

Security Assessment

Supernova

CertiK Verified on Dec 8th, 2022

CertiK Verified on Dec 8th, 2022

Supernova

The security assessment was prepared by CertiK, the leader in Web3.0 security.

Executive Summary

TYPES	ECOSYSTEM	METHODS
Chain	CosmosSDK	Manual Review, Static Analysis
LANGUAGE	**TIMELINE**	**KEY COMPONENTS**
Golang	Delivered on 12/08/2022	N/A

CODEBASE
https://github.com/Carina-labs/nova/
...View All

COMMITS
932b23ea391d4c89525c648e4103a3d6ee4531d5
...View All

Vulnerability Summary

34	31	0	0	3	0	0
Total Findings	Resolved	Mitigated	Partially Resolved	Acknowledged	Declined	Unresolved

0	Critical		Critical risks are those that impact the safe functioning of a platform and must be addressed before launch. Users should not invest in any project with outstanding critical risks.
6	Major	6 Resolved	Major risks can include centralization issues and logical errors. Under specific circumstances, these major risks can lead to loss of funds and/or control of the project.
6	Medium	6 Resolved	Medium risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform.
10	Minor	8 Resolved, 2 Acknowledged	Minor risks can be any of the above, but on a smaller scale. They generally do not compromise the overall integrity of the project, but they may be less efficient than other solutions.
12	Informational	11 Resolved, 1 Acknowledged	Informational errors are often recommendations to improve the style of the code or certain operations to fall within industry best practices. They usually do not affect the overall functioning of the code.

TABLE OF CONTENTS | SUPERNOVA

Summary

Executive Summary

Vulnerability Summary

Codebase

Audit Scope

Approach & Methods

Review Notes

System Overview

Modules

Findings

GLOBAL-01 : Missing Query Client Commands

932-01 : Proposal Handler in `poolincentive` Module

APP-01 : Potential Dead Code

CLA-01 : Improper Usage of `panic()`

DEP-01 : Incorrect Store Key Naming

GOV-01 : Missing `weight` Update

HOK-01 : Incorrectly Stored Data

HOK-02 : Discussion on `AfterTransferFail()`

MOP-01 : `gRPC` Services Not Registered

MOP-02 : Discussion on Module `poolincentive`

MSE-01 : Missing Save Data

MSE-02 : Incorrect Account Used When Withdraw

MSE-03 : Missing State Update

MSE-04 : Incorrect Withdraw Process

MSR-01 : Lack of Unique Check for `BaseDenom`

MST-01 : Incorrect Error Message

MSV-01 : Lack of Input Validation

QUE-01 : Incorrect Query Response

SEN-01 : Using Local Time

X93-01 : Missing Basic Validation

X93-02 : Missing Messages Codec Registration

X93-03 : Improved Address Validation

GLOBAL-02 : Discussion on `query.proto`

GLOBAL-03 : Discussion on `handler.go`

932-02 : Unused Variables and Consts

932-03 : Redundant Alias

ANT-01 : Unused Functions

GAL-01 : Typo

IBM-01 : Typo in File Name

MOU-01 : Duplicate Code

MSR-02 : Missing Emit Events

MSR-03 : Wrong Comments

MSR-04 : Discussion on Message `MsgChangeRegisteredZone` in Module `icacontrol`

TXL-01 : Unused Input Arguments

Optimizations

X93-04 : Improper Validation Sequence

Appendix

Disclaimer

CODEBASE | SUPERNOVA

Repository

https://github.com/Carina-labs/nova/

Commit

932b23ea391d4c89525c648e4103a3d6ee4531d5

AUDIT SCOPE | SUPERNOVA

133 files audited ● 4 files with Acknowledged findings ● 29 files with Resolved findings ● 100 files without findings

ID	File	SHA256 Checksum
APP	app/app.go	2949cf2c5020aafcb7adff3330050c67f86c3086d742e159b88bc77f34c1a0ad
MST	x/gal/types/msgs.go	385b402371d4a6b521c19b6262a04511b6f9ab8b421594c79dee3a100478f035
MSY	x/icacontrol/types/msgs.go	b61b0eb536aa48f4bfc9541b6cbff6372181b87502fe68d1a9a1c15ec7bf7398
MSO	x/poolincentive/types/msgs.go	e3940c115290b4271805399f5a689dac41ddd33c94f4a449ce1a0b625fac7c46
ANT	app/ante.go	630eaf6cae702e7fbbb93fa116af7195f68d0ad9192a76b5765cf3aa98a752be
MOU	app/keepers/modules.go	d686c954dffcf80989829ccafe3b2b21fd0622db58b8946b514f51f7924df944
ALI	x/airdrop/alias.go	750612eb23012751b458b60797f4068ee9c3b46d451c37c2c03aa20f04a4c2e8
GRP	x/airdrop/keeper/grpc_query.go	9fae2d5c44a0414c90c95f179a849cd2e3e92b811aa6ec594385aca3bdcca75c
MSG	x/airdrop/keeper/msg_server.go	60592e00a249138540596a41d89a711b957be0b58c4fee75dfbe55883e501130
TXL	x/gal/client/cli/tx.go	d98c87e0c1d277bdc6ca510029dd47adc17e72cfe80ae879af88bf8afff9e96e
CLA	x/gal/keeper/claim.go	5968458f3609200dce8b7edfe7cf345fa7f3cef943abd849d6737e47bffe98ee
DEP	x/gal/keeper/deposit.go	43e9b95f71c56c1c418b3a425518d6189a1f4c488cb613a7711ad2a5c64608f4
HOK	x/gal/keeper/hooks.go	3c505136348dac025b5c8b889baaba828505652772fba06561143279788d574a
MSE	x/gal/keeper/msg_server.go	510c6fa5a94372a4c395dbde0aef0f9c4a3873401f479dfcfb120bf4b687edf5

ID	File	SHA256 Checksum
COE	x/gal/types/codec.go	501b75777579295b794a1a577b20bd1d3ba87c0938173bb67eb6f08ac7af325c
ERO	x/gal/types/errors.go	5ef7db6667019ee241a2957c2587da0291882e70bf3a7c23178fe32cf584a739
ALS	x/icacontrol/alias.go	d3f414f5c0cd32cdca7df0d062807c937dd96ecf2b435bd6c4e724664d5cc797
IBM	x/icacontrol/ibc_mobule.go	57f55312dbf4aec12e6217003f4b944397bf4643f93818a68d29d154612203ff
GRQ	x/icacontrol/keeper/grpc_query.go	4ecdf1c577ef5d271676bd87b391b6535dfd89f7851a968cddc2fea2768695f1
IBH	x/icacontrol/keeper/ibc_handler.go	88795405a34e5ceacdcb4275070010bc11e94330a1e8a37bf92ecb82aafef5bf
MSR	x/icacontrol/keeper/msg_server.go	23b7ce553c4e0e611ef74184099f35f1d23a83cbad5d3f9eb8255f040757b0e7
SEN	x/icacontrol/keeper/send_msgs.go	215d8e9b8b295ad3a5d3b1e3655a7ea8d16b1030aa06adeab45aed1c48ea078a
ERS	x/icacontrol/types/errors.go	b92dd8990e2a4a7a47e94579df800f658dfb666dfa0bc5768256740798edb283
PAT	x/mint/types/params.go	6ad06c35a14cfb578377e220dfe9b92e80911b32210f1ccffce33b81a4e6bf82
ALO	x/oracle/alias.go	6126b8c48c45a5f81aace3a7a9451b648ff47be726f46ed3d884ad6019ae27ac
MSV	x/oracle/keeper/msg_server.go	3890e4b1d06c6b855a7489ad53b53f5319039b4f35536ecd0e5b5bf0b7cf4823
MSP	x/oracle/types/msgs.go	650ff471afbbba9c277355955a9132950eacfefbcf5362b33aebafc6847ec176
GOV	x/poolincentive/keeper/gov.go	29fabc4da33c137b0873bb0f3f6978e1acee425d462dc7615a461d15d297715b
COS	x/poolincentive/types/codec.go	d7accb39132d89b96d307713da194a351bae072c2093964e2e4cf7d8f7d84dee
GEL	x/poolincentive/types/genesis.go	d44bd650c00555c4d36370622cc3dc85ed20ba685d6adb68d3ed4a1cbb795f1b

ID	File	SHA256 Checksum
GOT	x/poolincentive/types/gov.go	7dc92762a6714b33582b9008f9a3a6beb07483e2a591b4fe910a1dc236b75f74
HAD	x/poolincentive/handler.go	bd73cbbb6b4f34f4bdf68656da7d090370855eb58f746bbf7f99dc320ac248aa
MOP	x/poolincentive/module.go	6501831569522e264f1f91b17c422f51ca565e62c9db9e2e29eb5e03e94c444d
EXP	app/export.go	89682754d1579e700dc5b52e3a4331092618251efff0ef8de14a38d8e69cdfc7
GEN	app/genesis.go	fa720055e77331d79edf6ea877ea4cc3ab4cbee282681b10f77143e77f82b2c4
MOD	app/modules.go	7483b4ebc37ade02409ea94463b1f0f5eb200197bf9e728187508a7bcb2d8e30
WAS	app/wasm_config.go	36c43bf4ba02d3873e78427ac9c4c699b160a922b69e939c4d46f1cd4a1eda5e
KEP	app/keepers/keepers.go	6d009b792db46f174f986af31428e0d0a0bc18ac878d66cb216b3ce5df5622b7
KEY	app/keepers/keys.go	891820d090ff5c59be81aa1af7174ebd3c8d80845c849e1f1ed82ccaa3566b1d
CON	app/params/config.go	0864be519bb3b020bdb388ecd0ba080bd7def2cf35abf815792273c23f0fc1c2
CMD	x/airdrop/client/cli/cmd.go	f643039b4610935a1fef84c1783100602094c5e80a1efdd3dd0058a0986165ea
QUY	x/airdrop/client/cli/query.go	6e36317656cdc34b842bb42cc0613b022ea3cb81548b9e7672275fdf2a621784
TXC	x/airdrop/client/cli/tx.go	2695bca142beda15dbe53d339e714ddc4fbed205bbcede47395fc5135f0179a9
MOL	x/airdrop/module.go	59725cf0fd32251deb2af05b848fafaebec4859515a8f7a46349aa6e23d79fb0
ACT	x/airdrop/keeper/action.go	cf01c45e2edf0ea10d9700bdb0eac11a017ddb2fea3bc28d00ea4073610e276e
AIR	x/airdrop/keeper/airdrop_info.go	902c50ee62d9a9b9e1cdc32f43e7da5d528dc0818be22fdf20623a29d57505d6
GEE	x/airdrop/keeper/genesis.go	c7a3b903196ac3d743506c30142b1ec03b5fd2aa958c0d4fcfedfd2f4b4baabe

ID	File	SHA256 Checksum
HOO	x/airdrop/keeper/hooks.go	dfbfd1665eeba44c7733c17c18ba48419be366012f847a34fec7052a0afbe2f5
INV	x/airdrop/keeper/invariants.go	3965cc60f35c0f79008edff950ea23b036090310ed148c58780b1f32708bbbb9
KER	x/airdrop/keeper/keeper.go	7b680a60a13b551a47f2d7ef6ecf53248693c9ddcfc5efc3571497a28f6a59ad
USE	x/airdrop/keeper/user_state.go	4722727e56283a1f1cb44d02539f6b81b62aedf892e92e997d147195b3e3400c
COD	x/airdrop/types/codec.go	43a1cd0e8cc9ead84b1e4a1017228a022d3525b3e6fd3bbb1d4b95525c411579
ERR	x/airdrop/types/errors.go	0f3248d46b65c825d302b499d21ae34ceee3d4a7eb461e02ac36c4a5f02e7b12
EXE	x/airdrop/types/expected_keepers.go	1873fbcb4474246dc901694d7e87fe1c2c349e8c8624c9f17c97263125d17e71
GES	x/airdrop/types/genesis.go	5ff8149be9647a06467204ee43eca46e9aaf67057f178ddad3a825f20e338a04
KET	x/airdrop/types/key.go	d53f09340c5b4784cadbb5534bfc3f9f2313c842925381d58fee4e2f28990e29
MSS	x/airdrop/types/msgs.go	3efee0963aa24ea745da07420a5f34115e4e1f25687883ce51f28c25baecdbf5
QUE	x/airdrop/types/querier.go	7c347886dbeed39a02f9f23d860ffb46fa1da70151c2268a6289325c55acf415
CMC	x/gal/client/cli/cmd.go	7b792f64d6d8744fb5fcfe33d812ca58e9b0934e5ce9b670d45eae7252ca0795
QUC	x/gal/client/cli/query.go	2f2c0719627e91e67f7e608a6f23731d725c33b5f0130712d9c7950e43625f55
DEL	x/gal/keeper/delegation.go	8c360aa56ceff39f7fb67dbdc6c33653092fb2167c9eacd5bca34c16218ff604
GRC	x/gal/keeper/grpc_query.go	70e6c1e7d42fcfbbd850ce6905babbc5b352477741d386c910094c807e6eef10
IBC	x/gal/keeper/ibc_transfer.go	44f5e5d7e279773267cece30e2bd6d2c8fcaf48e3da64f33010b1c1f88a3286e
INA	x/gal/keeper/invariants.go	45aaa6f20e8f6d11f95b3790cfe8bc35c1d66f1b5e438ed574c04218530e3347

ID	File	SHA256 Checksum
KEK	x/gal/keeper/keeper.go	6d10ab41511f4223405be4536ed104d24d241cd4e05c663dba395fbf5b079fa9
QUR	x/gal/keeper/querier.go	f50a32c2229f269ae10daeacea524eb4172f64693020d7b272118aa5c13d170b
UND	x/gal/keeper/undelegate.go	fe250087676c1c625da90639e9592ad82804e2c5388ea9731ab69e22f0020d31
WIT	x/gal/keeper/withdraw.go	c97c53931979a89a10057b96563ce043841029ca885aa7a0edcc36720fa1097d
EVE	x/gal/types/event.go	07a3c5955affb2d0d831ce22949f2fa797634d091d451d9b144088d17238bedd
EXC	x/gal/types/expected_keepers.go	bb2a8a95a995f3cd64535085d17365371715a362a76cad215c305057f62e8e5b
GEI	x/gal/types/genesis.go	5a69a98acb590abe9edc08205db1b38e1dae3b8391c178513ac18a093c16c90d
KES	x/gal/types/key.go	1a478ab22b12bbd24111927d85e1b4e129a97649edffb912de3f15f5e32299a1
PAA	x/gal/types/params.go	a0e048cf9050eff3164f9bacea4903ab1c872dd88a77ecea6b0177c73981342a
STA	x/gal/types/state.go	1e26e5dc6e1fb4cea313facabb9fa73e438ac570cab0437793aecb209a961c49
ALA	x/gal/alias.go	058afb85d3948bbab2bddb31de281b21defe0b7aac6b152fb36a88dc1fdbe48f
GEG	x/gal/genesis.go	6a2ae0e8677d5bc84ac17f969644fca90cf480a82abacc508c069e3d22435e29
MOE	x/gal/module.go	56d86b2c57448d1e378ea41c8bb0191c9d188c57bff096869e842ca1d763ec52
MOI	x/icacontrol/module.go	1837d4319566870c6064d85b640130e8e2e3cc1d6700cd00df7ed1d50280b55a
CML	x/icacontrol/client/cli/cmd.go	bc6b86c6a1380306597def56cb9aa0a522e46839961794df0c25a1013b5b1728
QUL	x/icacontrol/client/cli/query.go	2aa9d97edcb1d84695a2393743d4de0de6d0e647e0ec4a78ae174dbc3520b103
TXI	x/icacontrol/client/cli/tx.go	6d578d8be6d63f066f0be32ae1f71f4dcdbf6936dc13240edfc99f2d614935b4

ID	File	SHA256 Checksum
COT	x/icacontrol/keeper/controller_address.go	82a3a0c6d322dad295caf7afa3dc9c1c6bd8c9ebafe4727d2ded37fe50402e45
GEK	x/icacontrol/keeper/genesis.go	1fb9d9ac305d462053c25b3cd44b3e1a56dd356d82adf12c544c34dba325544f
HOS	x/icacontrol/keeper/hooks.go	1291209218bf1b91d9e1373a51eaf702a4645155c0f5e6275252d15662b46175
KEI	x/icacontrol/keeper/keeper.go	f7c5fa1c84e502d494151e1c55f940870528e23098b6ab60ac4220029b49e161
PAM	x/icacontrol/keeper/params.go	e888bff7eb665e23a1adee3c758f38931764c13456ae69d6aa5ec41b3c13869f
UTI	x/icacontrol/keeper/utils.go	a4103f75cf9068f2ccd02f103ffe71a14aa51d22b8909bd60593932415079288
VER	x/icacontrol/keeper/version.go	b88cf03ecca0cb5dabdc28d6e2c9f529cde3cf90a7dc52400a706be90d021eaa
ZON	x/icacontrol/keeper/zone.go	457d23a37748601f44eb97eaa68393684c9d901c4f97cc1734582f56f1de70fb
COC	x/icacontrol/types/codec.go	94c8ff9c510ac0ae9489f7d9e9a42d9d02c1e4ee5142677776414e7df9bff541
EXT	x/icacontrol/types/expected_keeper.go	cc2a2aef30bfc6c3b5cb1bd7cb7a279c3fc35ce71a87bc2ebd07127a2815fafb
GET	x/icacontrol/types/genesis.go	943a293646dfa504aa3a1bc51869bfd0b704696da37bbbc014c7c88bad698f7b
HOT	x/icacontrol/types/hooks.go	9464413d823c2156b7e6a972240622082489a239bfb1c921915a9fa17395d60f
KEC	x/icacontrol/types/keys.go	f2d07b1fa95fa9a39544bfc845fb7789aa0ab4e08d163dc8f3be1483e5f85e8a
PAS	x/icacontrol/types/params.go	92579ff1eba537f40496dee98cee85e758a1360ca485edfdffc8b01891d8e450
QUT	x/icacontrol/types/query.go	7c347886dbeed39a02f9f23d860ffb46fa1da70151c2268a6289325c55acf415
STT	x/icacontrol/types/state.go	ae0be109ab61bc8620d3d9bc203fbd1e39cbafd425bc33ee80df9045055784e1
GEM	x/mint/genesis.go	715452e1ee72c8bd2fd7f593f5d2f795a66293be51f98f1cd3314ef79d04015c

ID	File	SHA256 Checksum
MOM	x/mint/module.go	0bf74fc03bef689fb0f621465bf02472b078b7cbb3101893052528733a4e720a
ABC	x/mint/keeper/abci.go	8eb2395525c2be5033dddb6ab0d0096da1c7fde57d4d51d93395ccb0f9a415a3
GRU	x/mint/keeper/grpc_query.go	188a1ded78fde14e01ccf2a281c901bd5a19a11949154a6711e91fa6cee16919
KEM	x/mint/keeper/keeper.go	ed60c5e37c8c5540791ce114be19c65fd33d1210d560d8b35e763d0645bb6f3d
QUI	x/mint/keeper/querier.go	0a253bb20f519ecb795745d1d1ede916c938cd7ff2096b2cf2d1aed06b91aa0d
COY	x/mint/types/codec.go	e7f2bed92b9c0d4eb4044ec520190ae5f744d4efca3d5c09d1dcbe95b50bac19
EXD	x/mint/types/expected_keepers.go	f6a30b2a8337dd6de7226ea614b8b6c9670f822c6e47b961424ee21e5c8ef116
GEY	x/mint/types/genesis.go	12b1b1ec7aaae14372fdb6384167fb2333f4f2d058fcb7252f61769464f9ce3d
KEN	x/mint/types/keys.go	31b1349b5f976255e1e4491eab68e77bc4c860c327dd950feb69956898b3fe09
MIN	x/mint/types/minter.go	0f28c411afa110c91bd479aad50896c64589fd2674036d5583c60f62a2fd4c5c
QUN	x/oracle/client/cli/query.go	db21fa91167973369e8e05d9d7fcdbef992bfdc9ac313893f37269b44e360335
TXE	x/oracle/client/cli/tx.go	94b9f7d84a2f8d012b190514f4a8d47f69e67ea315beac49c7200ebfa37f713f
HAN	x/oracle/handler.go	09b0891633e6672a06c02551f2651f0ec049a0a8e860986a6eef618708ea6fff
MOO	x/oracle/module.go	11f3a3332f3dba1ec41c5fd0d28fe026dc8aa8ca7fccfa2c3c595c7179bc579f
GEP	x/oracle/keeper/genesis.go	d8aa27fa55753375e48e0774725e2cbdb8655298351b4724e1ffc88f12f42389
GRE	x/oracle/keeper/grpc_query.go	a9dc6c48f6e6e38305d310c190138f2b7613e5823c2385d8be24178b1badd21d
KEO	x/oracle/keeper/keeper.go	48a330dab05010aaff79905d4d4e14399cef177a2a76f34895a601440097cba1

ID	File	SHA256 Checksum
ORA	x/oracle/keeper/oracle_address.go	1e788a96ac3103f6f9ea62bebea4bc81bfebc441856d2abfc315f2a9b35b8037
PAK	x/oracle/keeper/params.go	60786ebf3654ad94394720716a2abacd2da77dde39d21daab0ca8ead0eab87a2
COP	x/oracle/types/codec.go	23deb52322e90c43d253ddcf696fdd44bea99cf2cb3319de1818739d6edf26fd
ERT	x/oracle/types/errors.go	da5fab1a9725f596cdb8357753b486a1ca4027a32309a66317eee5e57b6164aa
GEO	x/oracle/types/genesis.go	926180c8caad432d9500812a3962b0b92a41f053f5cbf745132289293b6102be
KEA	x/oracle/types/key.go	c28775aeb7660dacd96d72b551c9f8c89924640da5be43b94e2e2c99b62957b3
PAY	x/oracle/types/params.go	ffe831fc9196f0da7275d518b9bf86d78868fc902623eda4fbbf497ccec96906
QUP	x/oracle/types/querier.go	7971604e34d5cd8145ade1e955e25e059e516a4a26dd263073c9dd2a30f36135
QUO	x/poolincentive/client/cli/query.go	a3cac6e940a2b06b8ee72fb851bebf348d232554bbf5f55f06951bb6c7f4906b
TXN	x/poolincentive/client/cli/tx.go	aea8614196181824f59a29a7191b22bac56f5f426048a77de2fb0f5e5443b4df
GER	x/poolincentive/keeper/genesis.go	d3d3a2a24025aca5991aae7e2e9f19ff8d87ec341ea7d75a941d183f6dc6e698
HOE	x/poolincentive/keeper/hooks.go	f50a32c2229f269ae10daeacea524eb4172f64693020d7b272118aa5c13d170b
KEL	x/poolincentive/keeper/keeper.go	79ccbd3d2a01fb76ffb08d34c209afe0c1eab6b9b80b9558681111c9e99c8f25
PAE	x/poolincentive/keeper/params.go	986ed80d95eaea9d51fb30a96321f4e0cc8def3fd1efcce22529532cb7db4356
POO	x/poolincentive/keeper/pool.go	74b98a44c686f3a32ebd54ab5b988fa720c01cfa8526363f31274c8e55a05662
KEV	x/poolincentive/types/key.go	55919287f74aeda1c2ebaefb6ed0bc6e38a11461953922faa3bd051ecf50364a
PAP	x/poolincentive/types/params.go	f9b68d867fdcd8d3bd7f86fb5f4c73779607918db08e8e26e5b3e2af80a77ebc

ID	File	SHA256 Checksum
QUS	x/poolincentive/types/query.go	3ccb0a0467457719a923b7e1eea200e8dc17a802a0cc7a1f267ace55216cf7e6

APPROACH & METHODS | SUPERNOVA

This report has been prepared for Supernova to discover issues and vulnerabilities in the source code of the Supernova project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Manual Review and Static Analysis techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

The security assessment resulted in findings that ranged from critical to informational. We recommend addressing these findings to ensure a high level of security standards and industry practices. We suggest recommendations that could better serve the project from the security perspective:

- Testing the smart contracts against both common and uncommon attack vectors;
- Enhance general coding practices for better structures of source codes;
- Add enough unit tests to cover the possible use cases;
- Provide more comments per each function for readability, especially contracts that are verified in public;
- Provide more transparency on privileged activities once the protocol is live.

REVIEW NOTES | SUPERNOVA

System Overview

Supernova is a liquid staking platform for the cosmos ecosystem. Using IBC and ICA, tokens from multiple app chains in the Cosmos ecosystem can be staked and equity tokens can be minted. In addition, Supernova can securely trade liquidated assets through a decentralized exchange that allows you to trade tokens that match equity tokens.

Modules

Supernova is an App-Chain based on Cosmos-SDK. It consists of the following modules for smooth liquid staking.

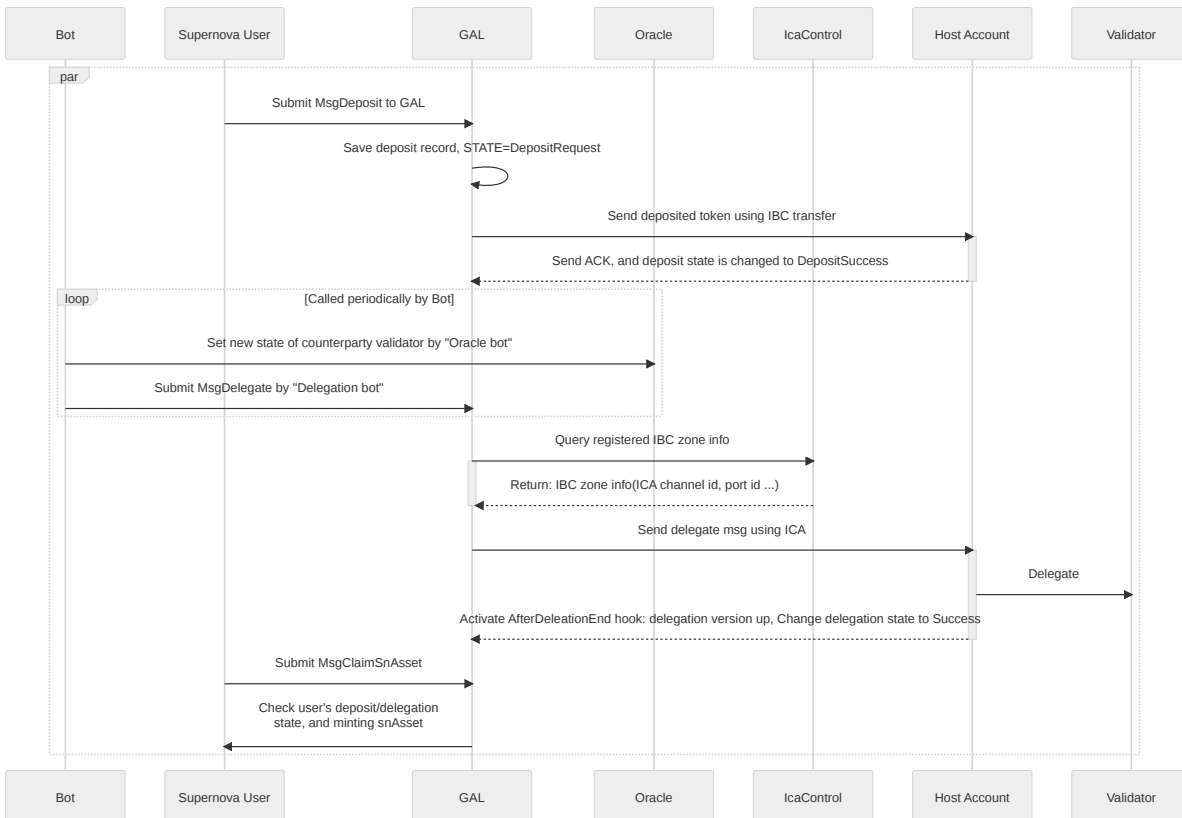
- GAL - The GAL module manages deposit records, undelegation and withdrawal records of users who want to use liquid staking.
- IcaControl - The IcaControl module manages the Interchain Account (ICA) required to ensure accurate operation of the liquid stacking.
- Oracle - The Oracle module manages the status (total delegation) of the validator of the counterpart zone to be delegated by Supernova. The reason why this information is needed is to calculate the equity when issuing equity tokens(snAsset).
- Mint - The Mint modules are responsible for minting and distributing Supernova's governance coin, Nova.
- Pool-Incentive - The Pool-Incentive module manages information to provide incentives to Supernova's liquidity providers.
- Airdrop - The Airdrop module is a module that manages information to incentivize early participants in Supernova.

DIAGRAMS | SUPERNOVA

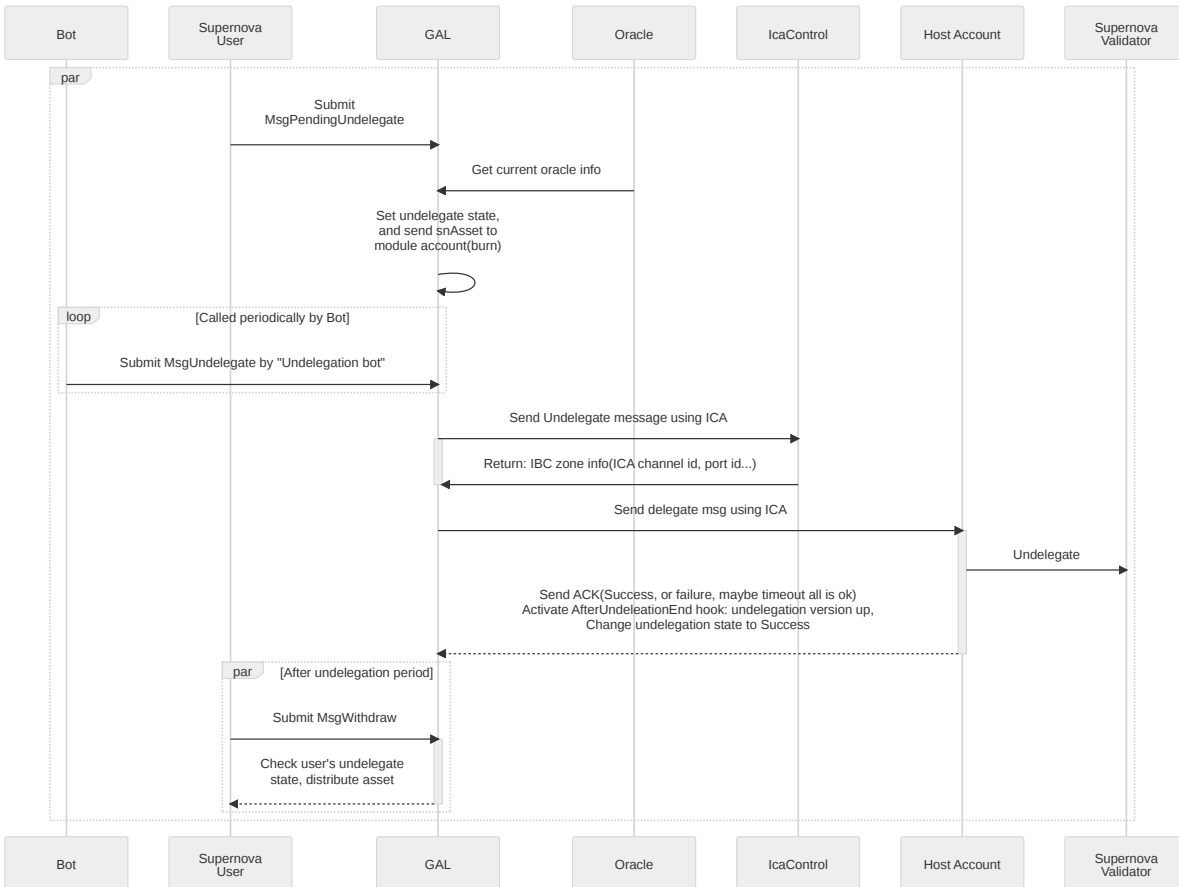
Instructions Sequence

The chart below depicts the flow of tokens deposited/deposited by each user to their assets.

Deposit Flow



Undelegate Flow


Bot
Supernova User
GAL
Oracle
IcaControl
Host Account
Supernova Validator
par
Submit MsgPendingUndelegate
Get current oracle info
Set undelegate state, and send snAsset to module account(burn)
loop
[Called periodically by Bot]
Submit MsgUndelegate by "Undelegation bot"
Send Undelegate message using ICA
Return: IBC zone info(ICA channel id, port id...)
Send delegate msg using ICA
Undelegate
Send ACK(Success, or failure, maybe timeout all is ok)
Activate AfterUndeleationEnd hook: undelegation version up,
Change undelegation state to Success
par
[After undelegation period]
Submit MsgWithdraw
Check user's undelegate state, distribute asset
Bot
Supernova User
GAL
Oracle
IcaControl
Host Account
Supernova Validator

FINDINGS | SUPERNOVA

34	0	6	6	10	12
Total Findings	Critical	Major	Medium	Minor	Informational

This report has been prepared to discover issues and vulnerabilities for Supernova. Through this audit, we have uncovered 34 issues ranging from different severity levels. Utilizing the techniques of Manual Review & Static Analysis to complement rigorous manual code reviews, we discovered the following findings:

ID	Title	Category	Severity	Status
GLOBAL-01	Missing Query Client Commands	Logical Issue	Minor	Resolved
932-01	Proposal Handler In `poolincentive` Module	Logical Issue	Major	Resolved
APP-01	Potential Dead Code	Logical Issue	Minor	Acknowledged
CLA-01	Improper Usage Of `panic()`	Volatile Code	Minor	Resolved
DEP-01	Incorrect Store Key Naming	Logical Issue	Minor	Resolved
GOV-01	Missing `weight` Update	Logical Issue	Major	Resolved
HOK-01	Incorrectly Stored Data	Logical Issue	Major	Resolved
HOK-02	Discussion On `AfterTransferFail()`	Logical Issue	Minor	Resolved
MOP-01	`gRPC` Services Not Registered	Logical Issue	Medium	Resolved
MOP-02	Discussion On Module `poolincentive`	Logical Issue	Minor	Resolved

ID	Title	Category	Severity	Status
MSE-01	Missing Save Data	Logical Issue	Major	Resolved
MSE-02	Incorrect Account Used When Withdraw	Logical Issue	Major	Resolved
MSE-03	Missing State Update	Logical Issue	Major	Resolved
MSE-04	Incorrect Withdraw Process	Logical Issue	Medium	Resolved
MSR-01	Lack Of Unique Check For `BaseDenom`	Volatile Code	Medium	Resolved
MST-01	Incorrect Error Message	Logical Issue	Minor	Resolved
MSV-01	Lack Of Input Validation	Volatile Code	Minor	Resolved
QUE-01	Incorrect Query Response	Volatile Code	Medium	Resolved
SEN-01	Using Local Time	Volatile Code	Minor	Resolved
X93-01	Missing Basic Validation	Volatile Code	Medium	Resolved
X93-02	Missing Messages Codec Registration	Logical Issue	Medium	Resolved
X93-03	Improved Address Validation	Volatile Code	Minor	Acknowledged
GLOBAL-02	Discussion On `query.proto`	Language Specific	Informational	Acknowledged
GLOBAL-03	Discussion On `handler.go`	Volatile Code	Informational	Resolved
932-02	Unused Variables And Consts	Coding Style	Informational	Resolved

ID	Title	Category	Severity	Status
932-03	Redundant Alias	Coding Style	Informational	Resolved
ANT-01	Unused Functions	Coding Style	Informational	Resolved
GAL-01	Typo	Coding Style	Informational	Resolved
IBM-01	Typo In File Name	Coding Style	Informational	Resolved
MOU-01	Duplicate Code	Coding Style	Informational	Resolved
MSR-02	Missing Emit Events	Coding Style	Informational	Resolved
MSR-03	Wrong Comments	Inconsistency	Informational	Resolved
MSR-04	Discussion On Message `MsgChangeRegisteredZone` In Module `icacontrol`	Volatile Code	Informational	Resolved
TXL-01	Unused Input Arguments	Coding Style	Informational	Resolved

GLOBAL-01 | MISSING QUERY CLIENT COMMANDS

Category	Severity	Location	Status
Logical Issue	Minor		Resolved

Description

The following `gRPC` queries in module `gal` are not registered to client commands:

- rpc EstimateSnAsset(QueryEstimateSnAssetRequest) returns (QueryEstimateSnAssetResponse)
- rpc DepositAmount(QueryDepositAmountRequest) returns (QueryDepositAmountResponse)
- rpc DelegateCurrentVersion(QueryCurrentDelegateVersion) returns (QueryCurrentDelegateVersionResponse)
- rpc UndelegateCurrentVersion(QueryCurrentUndelegateVersion) returns (QueryCurrentUndelegateVersionResponse)
- rpc WithdrawCurrentVersion(QueryCurrentWithdrawVersion) returns (QueryCurrentWithdrawVersionResponse)

Recommendation

We recommend adding these queries to command.

Alleviation

`[CertiK]` : Supernova team heeded the advice and resolved this finding in commit cf6357f3d811af1de05207d0c490b74a2a65c698.

932-01 | PROPOSAL HANDLER IN `poolincentive` MODULE

Category	Severity	Location	Status
Logical Issue	Major	app/keepers/modules.go: 46~56; x/poolincentive/handler.go: 11~22	Resolved

Description

The proposal handler will be created by the function `NewPoolIncentivesProposalHandler()`, but the proposal handlers are not declared in `client` folder.

You can refer to the built-in module `params` :

1. Create the proposal handler by function `NewParamChangeProposalHandler()` :

   https://github.com/cosmos/cosmos-sdk/blob/78886bc8de55b391a44b3bf28b617c60f173fd80/x/params/proposal_handler.go#L13-L24

2. Declaration of proposal handler in `client` folder :

   https://github.com/cosmos/cosmos-sdk/blob/78886bc8de55b391a44b3bf28b617c60f173fd80/x/params/client/proposal_handler.go#L9

Also, we should append the proposal handlers to `gov` module when the basic module is generated(file : /app/keepers/modules.go) :

```
    gov.NewAppModuleBasic(
        append(
            wasmclient.ProposalHandlers,
            paramsclient.ProposalHandler,
            distrclient.ProposalHandler,
            upgradeclient.ProposalHandler,
            upgradeclient.CancelProposalHandler,
            ibcclientclient.UpdateClientProposalHandler,
            ibcclientclient.UpgradeProposalHandler,
        ),...,
    ),
```

Recommendation

We recommend adding the declaration of proposal handler in `client` folder in module `poolincentive`.

Alleviation

`[CertiK]` : Supernova team heeded the advice and resolved this finding in commit 734bacdfdcaf6fd2112f6781a33d16c1a8196329.

APP-01 | POTENTIAL DEAD CODE

Category	Severity	Location	Status
Logical Issue	Minor	app/app.go: 137~140	Acknowledged

Description

```
77  var (
78    // WasmProposalsEnabled enables all x/wasm proposals when it's value is "true"
79    // and EnableSpecificWasmProposals is empty. Otherwise, all x/wasm proposals
80    // are disabled.
81    WasmProposalsEnabled = "true"
82    // EnableSpecificWasmProposals, if set, must be comma-separated list of values
83    // that are all a subset of "EnableAllProposals", which takes precedence over
84    // WasmProposalsEnabled.
85    //
86    // See: https://github.com/CosmWasm/wasmd/blob/02a54d33ff2c064f3539ae12d75d027d9c665f05/x/wasm/internal/types/proposal.go#L28-L34
87
88    EnableSpecificWasmProposals = ""
89
90    EmptyWasmOpts []wasm.Option
91  )
```

According to the above statement, the variables `WasmProposalsEnabled` and `EnableSpecificWasmProposals` set their values only in the initial statement. They are used in the following statements. Their values are no more changed before they are used. Based on their default values, statements after line 142 will never be reached.

```
136 func GetWasmEnabledProposals() []wasm.ProposalType {
137   if EnableSpecificWasmProposals == "" {
138     if WasmProposalsEnabled == "true" {
139       return wasm.EnableAllProposals
140     }
141     ......
```

Recommendation

We recommend reviewing the logic to ensure it meets the design intent.

Alleviation

`[Supernova]` : Issue acknowledged. I won't make any changes for the current version.

CLA-01 | IMPROPER USAGE OF `panic()`

Category	Severity	Location	Status
Volatile Code	Minor	x/gal/keeper/claim.go: 28, 35	Resolved

Description

```
26 func precisionMultiplier(prec int64) *big.Int {
27   if prec > snAssetDecimal {
28     panic(fmt.Sprintf("too much precision, maximum %v, provided %v",
snAssetDecimal, prec))
29   }
30   return precisionMultipliers[prec]
31 }
```

The `panic()` function in Go Language is similar to exceptions raised at runtime when an error is encountered. `panic()` is either raised by the program itself when an unexpected error occurs or the programmer throws the exception on purpose for handling particular errors. And the `panic()` function is inbuilt into Go Language and when it is raised, the code prints a panic message, and the function crashes.

The functions `precisionMultiplier()` and `calcPrecisionMultiplier()` may crash the process when they raise errors. Maybe the team can use `fmt.Errorf()` here.

Recommendation

We recommend using `fmt.Errorf()` here.

Alleviation

`[CertiK]` : SuperNova team heeded the advice and resolved the finding in the commit hash `ab75fe92402996dba24c1eb3206a928a7727a72b` .

DEP-01 | INCORRECT STORE KEY NAMING

Category	Severity	Location	Status
Logical Issue	Minor	x/gal/keeper/deposit.go: 20, 26, 28, 64	Resolved

Description

In file `x/gal/keeper/deposit.go` , the naming of the key that is used to access the store of DepositRecords is incorrect.

```
18 func (k Keeper) SetDepositRecord(ctx sdk.Context, msg *types.DepositRecord) {
19     store := k.getDepositRecordStore(ctx)
20     key := msg.ZoneId + msg.Claimer
21     bz := k.cdc.MustMarshal(msg)
22     store.Set([]byte(key), bz)
23 }
```

```
26 func (k Keeper) GetUserDepositRecord(ctx sdk.Context, zoneId string, claimer
sdk.AccAddress) (result *types.DepositRecord, found bool) {
27   store := k.getDepositRecordStore(ctx)
28   key := []byte(zoneId + claimer.String())
29   if !store.Has(key) {
30     return nil, false
31   }
32
33   res := store.Get(key)
34   var record types.DepositRecord
35   k.cdc.MustUnmarshal(res, &record)
36   return &record, true
37 }
```

```go
func (k Keeper) GetTotalDepositAmtForZoneId(ctx sdk.Context, zoneId, denom string, state types.DepositStatusType) sdk.Coin {
  totalDepositAmt := sdk.Coin{
    Amount: sdk.NewIntFromUint64(0),
    Denom:  denom,
  }

  k.IterateDepositRecord(ctx, zoneId, func(index int64, depositRecord types.DepositRecord) (stop bool) {
    for _, record := range depositRecord.Records {
      if record.State == state && record.Amount.Denom == denom {
        totalDepositAmt = totalDepositAmt.Add(*record.Amount)
      }
    }
    return false
  })

  return totalDepositAmt
}
```

Recommendation

We recommend renaming put-in parameter `claimer` to `depositor`.

Alleviation

`[CertiK]`: Supernova team heeded the advice and resolved this finding in commit bbeedbc2ceb7b7ddf2f9ef5b2c800a92f619f546.

GOV-01 | MISSING `weight` UPDATE

Category	Severity	Location	Status
Logical Issue	Major	x/poolincentive/keeper/gov.go: 25~43	Resolved

Description

File : x/poolincentive/keeper/gov.go

When the proposal handle the message `UpdatePoolIncentivesProposal`, the method `HandleUpdatePoolIncentivesProposal()` will be executed to update the incentive pools. But there are only fields `PoolId` and `PoolContractAddress` updated, the fields `Weight` and `TotalWeight` are not updated.

Recommendation

We recommend adding the logic to update the `weight` of `IncentivePoolInfo`.

Alleviation

`[CertiK]`: Supernova team heeded the advice and resolved this finding in commit 8d5f8ffcc1c73298c2e22b740120731563daaa3d.

HOK-01 | INCORRECTLY STORED DATA

Category	Severity	Location	Status
Logical Issue	Major	x/gal/keeper/hooks.go: 193, 207, 224	Resolved

Description

```
185 func (h Hooks) AfterDelegateFail(ctx sdk.Context, delegateMsg
stakingtypes.MsgDelegate) {
186     zone := h.k.icaControlKeeper.GetRegisteredZoneForValidatorAddr(ctx,
delegateMsg.ValidatorAddress)
187
188     versionInfo := h.k.GetDelegateVersion(ctx, zone.ZoneId)
189     currentVersion := versionInfo.CurrentVersion
190
191     versionInfo.Record[currentVersion] = &types.IBCTrace{
192         Height:  uint64(ctx.BlockHeight()),
193         Version: types.IcaFail,
194     }
195
196     h.k.SetDelegateVersion(ctx, zone.ZoneId, versionInfo)
197 }
```

The functions `AfterDelegateFail()`, `AfterUndelegateFail()`, and `AfterTransferFail()` are used to handle failed requests. These functions should modify the `State` of the `versionInfo.Record` at the end of the process.

Recommendation

We recommend modifying the `State' value of each` versionInfo.Record' correctly.

Alleviation

`[CertiK]` : Supernova team heeded the advice and resolved this finding in commit 145d49a082c39e695bfb8bfca65fb5845a71af7a.

HOK-02 | DISCUSSION ON `AfterTransferFail()`

Category	Severity	Location	Status
Logical Issue	Minor	x/gal/keeper/hooks.go: 219	Resolved

Description

The function `AfterTransferFail()` is used to handle failed transfer requests. Why does it get `versionInfo` with the same logic as `AfterUndelegateFail()`? Please review this function to make sure it meets the design intent.

Recommendation

We recommend reviewing the logic to ensure it meets the design intent.

Alleviation

`[Supernova]` : This was incorrectly implemented logic, now fixed.

`[CertiK]` : Supernova team heeded the advice and resolved this finding in commit 3fad506d0066bd303f28b266c929a1c91f407281.

MOP-01 | gRPC SERVICES NOT REGISTERED

Category	Severity	Location	Status
Logical Issue	Medium	x/poolincentive/module.go: 130~132	Resolved

Description

The messages `CreateCandidatePool`, `SetPoolWeight`, `CreateIncentivePool`, and `SetMultiplePoolWeight` are declared in module `poolincentive`, but the `gRPC` service is not registered.

Also, there should have a function to implement the MsgServer interface for the keeper, and methods used for handling messages should be declared.

Recommendation

We recommend registering the gRPC service for messages and queries, and adding the function and methods we mentioned in description.

Alleviation

`[CertiK]`: SuperNova team fixed the issue in the commit hash `c95e81bdc0da8b847636da61d67f1afe00a1924a` on October 24th.

MOP-02 | DISCUSSION ON MODULE `poolincentive`

Category	Severity	Location	Status
Logical Issue	Minor	x/poolincentive/module.go: 1	Resolved

Description

In our opinion, the code in the following part of the module `poolincentive` is incomplete, please let us know if this part of the code is under development.

1. There is no `msg_server.go` for handling declared messages, and the is no gRPC service registered in `module.go`.

```
// RegisterServices registers a gRPC query service to respond to the
// module-specific gRPC queries.
func (am AppModule) RegisterServices(cfg module.Configurator) {
    // types.RegisterQueryServer(cfg.QueryServer(), am.keeper)
}
```

2. The handlers of proposal messages are not registered to client comments, and they also need to be appended into the basic module of `gov`.
3. In `x/poolincentive/types/msgs.go`, the methods `GetSignBytes` for each message are not implemented.

Recommendation

We recommend confirming that each service is registered prior to use.

Alleviation

`[CertiK]`: Supernova team heeded the advice and resolved this finding in commit `c444ac47cb5127c9fc8c35246cde0a1a14a437e4`

MSE-01 | MISSING SAVE DATA

Category	Severity	Location	Status
Logical Issue	Major	x/gal/keeper/msg_server.go: 138~141, 289~292, 410~413	Resolved

Description

```
138   versionInfo.Record[delegate.Version] = &types.IBCTrace{
139     Version: versionInfo.CurrentVersion,
140     State:   types.IcaRequest,
141   }
```

The variable `versionInfo.Record` has been modified to a new value, but has not been saved.

Recommendation

We recommend saving the new value with methd `set()` after the modification statement.

Alleviation

`[CertiK]` : SuperNova team fixed the issue in commit `145d49a082c39e695bfb8bfca65fb5845a71af7a` .

MSE-02 | INCORRECT ACCOUNT USED WHEN WITHDRAW

Category	Severity	Location	Status
Logical Issue	Major	x/gal/keeper/msg_server.go: 334	Resolved

Description

When users withdraw funds from this reserve, the tokens should be transferred from `moduleAccount` instead of `zoneInfo.IcaAccount.ControllerAddress`.

Recommendation

Review the relevant statement to ensure it meets design intent. If the contract should withdraw funds from `moduleAccount`, we recommend using `SendCoinsFromModuleToAccount()` function.

Alleviation

`[CertiK]`: SuperNove team heeded the advice and resolved the finding in the commit hash `db0d75850574bebbaf68b2f79eb2eb430386187c`.

MSE-03 | MISSING STATE UPDATE

Category	Severity	Location	Status
Logical Issue	Major	x/gal/keeper/msg_server.go: 405~408	Resolved

Description

File : x/gal/keeper/msg_server.go

In method `IcaWithdraw()` , if transaction failed, the state of version is not modified, it would cause a `WithdrawRecord` to be inaccessible and the corresponding funds will become un-withdrawn.

The reason will be follow :

1. There is a withdraw record which has valid version and the record state is `types.WithdrawStatusRegistered` . Also the state of the version is `types.IcaPending` .
2. As the state is `types.IcaPending` , the if branch in line 377 will be executed :

```
376     if version.State == types.IcaPending {
377     withdrawAmount = m.keeper.GetTotalWithdrawAmountForZoneId(ctx, msg.ZoneId, zoneInfo.BaseDenom, msg.ChainTime)
378   }
```

Method `GetTotalWithdrawAmountForZoneId()` in `x/gal/keeper/withdraw.go`

```
func (k Keeper) GetTotalWithdrawAmountForZoneId(ctx sdk.Context, zoneId, denom string, blockTime time.Time) sdk.Coin {
    amount := sdk.NewCoin(denom, sdk.ZeroInt())

    k.IterateWithdrawRecords(ctx, zoneId, func(index int64, withdrawInfo *types.WithdrawRecord) (stop bool) {
        for _, record := range withdrawInfo.Records {
            if record.CompletionTime.Before(blockTime) && record.State == types.WithdrawStatusRegistered {
                amount.Amount = amount.Amount.Add(record.Amount)
                record.State = types.WithdrawStatusTransferRequest
            }
        }
        k.SetWithdrawRecord(ctx, withdrawInfo)
        return false
    })
    return amount
}
```

We will find that if a withdraw record can be withdrawn, the state of withdraw record will be changed from `types.WithdrawStatusRegistered` to `types.WithdrawStatusTransferRequest`.

3. Then ibc message will be created and the transaction will be sent. If the transaction failed the method `IcaWithdraw` will return. And now the state of withdraw record is `types.WithdrawStatusTransferRequest` and the state of version is still `types.IcaPending`.
4. Then we execute method `IcaWithdraw()` again with the same parameters.
5. Because the state of version is still `types.IcaPending`, the if branch in line 377 will be executed again. But the state of record is `types.WithdrawStatusTransferRequest`, it means this record can't be withdrawn.

Recommendation

We recommend adding state modification when transaction failed.

Alleviation

`[CertiK]`: Supernova team heeded the advice and resolved this finding in commit bbeedbc2ceb7b7ddf2f9ef5b2c800a92f619f546.

MSE-04 | INCORRECT WITHDRAW PROCESS

Category	Severity	Location	Status
Logical Issue	Medium	x/gal/keeper/msg_server.go: 415~423	Resolved

Description

```
415 if err = ctx.EventManager().EmitTypedEvent(types.NewEventIcaWithdraw(
416     zoneInfo.IcaAccount.HostAddress,
417     zoneInfo.IcaAccount.ControllerAddress,
418     &withdrawAmount,
419     zoneInfo.IcaConnectionInfo.ConnectionId,
420     msg.IcaTransferChannelId,
421     msg.IcaTransferPortId)); err != nil {
422     return nil, err
423 }
```

In the function `IcaWithdraw()`, if the call to `ctx.EventManager().EmitTypedEvent(types.NewEventIcaWithdraw()` crashes, the `version.State` of the related record should be modified to `IcaFail`. Otherwise, the failed record may be blocked forever.

Recommendation

We recommend modifying `version.State` correctly.

Alleviation

`[CertiK]`: SuperNova team fixed the issue in the commit hash `145d49a082c39e695bfb8bfca65fb5845a71af7a` on October 24th.

MSR-01 | LACK OF UNIQUE CHECK FOR `BaseDenom`

Category	Severity	Location	Status
Volatile Code	Medium	x/icacontrol/keeper/msg_server.go: 32, 99	Resolved

Description

According to the logic of `GetsnDenomForBaseDenom()` function in `x/icacontrol/keeper/zone.go`, the `BaseDenom` variable should be unique.

Recommendation

We recommend adding unique check in the functions `RegisterZone()` and `ChangeRegisteredZone()` functions.

Alleviation

`[CertiK]` : Supernova team heeded the advice and resolved this finding in commit d35538678c1b24b784f9b02f53fc78a945fb04e4.

MST-01 | INCORRECT ERROR MESSAGE

Category	Severity	Location	Status
Logical Issue	Minor	x/gal/types/msgs.go: 311	Resolved

Description

```
310 if msg.ChainTime.IsZero() {
311     return sdkerrors.Wrap(ErrInvalidTime, msg.ControllerAddress)
312 }
```

The variable should be `msg.ChainTime` instead of `msg.ControllerAddress`.

Recommendation

We recommend correcting the error message to improve the code maintainability

Alleviation

`[CertiK]` : Supernova team heeded the advice and resolved this finding in commit 69edc70d5fd56cc07a4d04b3b04fc9ef7021e009.

MSV-01 | LACK OF INPUT VALIDATION

Category	Severity	Location	Status
Volatile Code	Minor	x/oracle/keeper/msg_server.go: 23	Resolved

Description

File : `x/oracle/keeper/msg_server.go`

In method `UpdateChainState()` , the following fields in input parameter `state *types.MsgUpdateChainState` are not validated.

- ZoneId : Passed-in `ZoneId` must be valid id for registered zone.
- Coin : Passed-in `Coin` must be registered token.

Recommendation

We recommend adding the validations for fields of input parameters.

Alleviation

`[CertiK]` : Supernova team heeded the advice and resolved this finding in commit e9a77001e4c95014f7498c49531061cc2c467e21.

QUE-01 | INCORRECT QUERY RESPONSE

Category	Severity	Location	Status
Volatile Code	Medium	proto/nova/poolincentive/v1/query.proto: 16	Resolved

Description

In `proto/nova/poolincentive/v1/query.proto`, the response of query `SingleCandidatePool` should be `QuerySingleCandidatePoolResponse`.

```
16   rpc SingleCandidatePool(QuerySingleCandidatePool) returns
(QuerySingleCandidatePool);
```

```
33 message QuerySingleCandidatePool {
34   string pool_id = 1;
35 }
36
37 message QuerySingleCandidatePoolResponse {
38   string pool_id = 1;
39   string pool_address = 2;
40 }
```

Recommendation

We recommend correcting the response of query `SingleCandidatePool` to `QuerySingleCandidatePoolResponse`.

Alleviation

`[CertiK]` : Supernova team heeded the advice and resolved this finding in commit d08f1d8bc6e594f2b4817231f1186b78f6cd7b70.

SEN-01 | USING LOCAL TIME

Category	Severity	Location	Status
Volatile Code	Minor	x/icacontrol/keeper/send_msgs.go: 42	Resolved

Description

When bot submits tx with `time.Now()` , it may cause of consensus error.

Recommendation

We recommend using `ctx.BlockTime()` to get timestamp on blockchain now.

Alleviation

`[CertiK]` : Supernova team heeded the advice and resolved this finding in commit 4c4bdac0dad50d0d695e74ae47f88f89be0542c3.

X93-01 | MISSING BASIC VALIDATION

Category	Severity	Location	Status
Volatile Code	Medium	x/icacontrol/types/msgs.go: 216, 261, 304; x/oracle/types/msgs.go: 30~33; x/poolincentive/types/genesis.go: 24~27; x/poolincentive/types/gov.go: 59~63, 104~108; x/poolincentive/types/msgs.go: 33~35, 62~68, 95~101, 127~133	Resolved

Description

The fields of messages that are never validated are listed blew :

In `x/icacontrol/types/msgs.go`

1. Fields `IcaTransferPortId` and `IcaTransferChannelId` in message `MsgIcaTransfer` are not validated.
2. Field `ZoneId` in message `MsgDeleteRegisteredZone` is not validated.
3. Field `ZoneId` in message `MsgChangeRegisteredZone` is not validated.

In `x/oracle/types/msgs.go`

1. Fields in message `MsgUpdateChainState` are not validated.

In `x/poolincentive/types/msgs.go`

1. All of the fields in message `MsgCreateCandidatePool` are not validated.
2. The fields `PoolId` and `NewWeight` in message `MsgSetPoolWeight` are not validated.
3. The fields `PoolId` and `PoolContractAddress` in message `MsgCreateIncentivePool` are not validated.
4. The slice field `NewPoolData` in message `MsgSetMultiplePoolWeight` is not validated.

In `x/poolincentive/types/genesis.go`

The method `ValidateBasic()` is not implemented, this will cause the validation of proposal messages `ReplacePoolIncentivesProposal` and `UpdatePoolIncentivesProposal` to be ineffective in `x/poolincentive/types/gov.go` .

`genesis.go`

```go
func (ip IncentivePool) ValidateBasic() error {
    // TODO : validate contract address is a valid cosm-wasm contract address.
    return nil
}
```

`gov.go`

```go
func (p *ReplacePoolIncentivesProposal) ValidateBasic() error {
    err := govtypes.ValidateAbstract(p)
    if err != nil {
        return err
    }
    if len(p.NewIncentives) == 0 {
        return fmt.Errorf("there is no incetive pool information")
    }

    for _, pool := range p.NewIncentives {
        if err := pool.ValidateBasic(); err != nil {
            return err
        }
    }

    return nil
}
```

```go
func (p *UpdatePoolIncentivesProposal) ValidateBasic() error {
    err := govtypes.ValidateAbstract(p)
    if err != nil {
        return err
    }
    if len(p.UpdatedIncentives) == 0 {
        return fmt.Errorf("there is no incetive pool information")
    }

    for _, incentive := range p.UpdatedIncentives {
        if err := incentive.ValidateBasic(); err != nil {
            return err
        }
    }

    return nil
}
```

Recommendation

We recommend adding validation for these fields in method `ValidateBasic()` to each message.

Alleviation

`[CertiK]` : Supernova team heeded the advice and resolved this finding in commit 954cda5cd6ce950af76989c43f68c2faf01ba33b and 54f49b4b332142d621ffd2f7b4f6fbbca808edd8.

X93-02 | MISSING MESSAGES CODEC REGISTRATION

Category	Severity	Location	Status
Logical Issue	Medium	x/gal/types/codec.go: 15~23; x/poolincentive/types/codec.go: 15~32	Resolved

Description

In the linked position, the codec and interface of `Msg` s have not been registered.

In Module `gal`

None of the messages are registered in file `x/gal/types/codec.go` :

```
var (
    amino     = codec.NewLegacyAmino()
    ModuleCdc = codec.NewAminoCodec(amino)
)

func RegisterLegacyAminoCodec(cdc *codec.LegacyAmino) {
}

func RegisterInterfaces(registry types.InterfaceRegistry) {
    msgservice.RegisterMsgServiceDesc(registry, &_Msg_serviceDesc)
    registry.RegisterImplementations(
        (*sdk.Msg)(nil),
    )
}
```

In Module `poolincentive`

The proposal `Msg` s `ReplacePoolIncentivesProposal` and `UpdatePoolIncentivesProposal` are not registered.

We can refer to the proposal `Msg` `in built-in module`

1. Declaration
2. Register codec
3. Register implementations

References:

1. Amino Documentation in Cosmos SDK

2. Messages : legacy-amino-legacymsgs

Recommendation

We recommend adding the codec registration of `Msg` services in modules.

Alleviation

`[CertiK]` : Supernova team heeded the advice and resolved this finding in commit c95e81bdc0da8b847636da61d67f1afe00a1924a.

X93-03 | IMPROVED ADDRESS VALIDATION

Category	Severity	Location	Status
Volatile Code	Minor	x/gal/types/msgs.go: 132~141; x/icacontrol/types/msgs.go: 50~77, 304~315, 357~372, 395~409; x/poolincentive/types/msgs.go: 95~101	Acknowledged

Description

The logic is that linked positions lack validations to ensure the passed-in addresses are valid :

In file `x/gal/types/msgs.go` :

- message `MsgUndelegate` : The field `ControllerAddress` lacks validation.

In file `x/icacontrol/types/msgs.go` :

- message `MsgRegisterZone` : The fields `IcaAccount.HostAddress` and `ValidatorAddress` lack validation.
- message `MsgChangeRegisteredZone` : The fields `IcaAccount.HostAddress` and `ValidatorAddress` lack validation.
- message `MsgIcaAuthzGrant` : The field `Grantee` lacks validation.
- message `MsgIcaAuthzRevoke` : The field `Grantee` lacks validation.

Recommendation

We recommend implementing a basic validation for addresses by function `AccAddressFromBech32()` .

Alleviation

Supernova team acknowledged this finding.

GLOBAL-02 | DISCUSSION ON `query.proto`

Category	Severity	Location	Status
Language Specific	Informational		Acknowledged

Description

We recommend the client use annotations in `query.proto` files to specify data conversion from HTTP/JSON to `gRPC` for queries. This is recommended in documentation Transcoding HTTP/JSON to gRPC.

For example, the query `Params` in module `gal` is assigned as below:

```
rpc Params(QueryParamsRequest) returns (QueryParamsResponse) {
  option (google.api.http).get = "/nova/gal/v1/params";
}
```

Recommendation

We recommend reviewing the logic to ensure it meets design intent.

Alleviation

Supernova team acknowledged this finding.

GLOBAL-03 | DISCUSSION ON `handler.go`

Category	Severity	Location	Status
Volatile Code	Informational		Resolved

Description

In our opinion, all of the `handler.go` files are removed from `Cosmos SDK` from version `v0.46`. According to the `go.mod` file the SDK Supernova used is v0.45.8, please tell us why did the team remove all of the `handler.go` files.

Reference:

1. Pull #9650 : remove legacy handler
2. Cosmos SDK CHANGELOG.md

Recommendation

We recommend reviewing the logic to ensure it meets the design intent.

Alleviation

`[CertiK]` : Supernova team heeded the advice and resolved this finding in commit 27b4edc17528d81e579427c90667f9a3b1de8ca4.

`[Supernova]` : The handler code was not written because the code was no longer used in the next version(v0.46). However, I added the code because I thought I should use a handler in versions earlier than v0.46.

932-02 | UNUSED VARIABLES AND CONSTS

Category	Severity	Location	Status
Coding Style	Informational	app/app.go: 90; x/airdrop/alias.go: 7, 8; x/gal/types/errors.go: 9~10, 20; x/icacontrol/alias.go: 6~7; x/icacontrol/types/errors.go: 8~9; x/oracle/alias.go: 7	Resolved

Description

The variables in the linked position are never used in Supernova.

/app/app.go

```
90      EmptyWasmOpts []wasm.Option
```

/x/airdrop/alias.go

```
7       StoreKey   = types.StoreKey
8       RouteKey   = types.RouterKey
```

/x/gal/types/errors.go

```
3       ErrCanNotReplaceRecord  = errors.Register(ModuleName, 3, "cannot replace
record")
4       ErrInsufficientFunds    = errors.Register(ModuleName, 4, "cannot withdraw
funds : insufficient fund")
```

```
14      ErrInvalidParameter     = errors.Register(ModuleName, 14, "invalid
parameter")
```

/x/icacontrol/types/errors.go

```
8       ErrIBCAccountAlreadyExist = sdkerrors.Register(ModuleName, 2, "interchain
account already registered")
9       ErrIBCAccountNotExist     = sdkerrors.Register(ModuleName, 3, "interchain
account not exist")
```

/x/icacontrol/alias.go

```
6       ModuleName = types.ModuleName
7       StoreKey   = types.StoreKey
```

/x/oracle/alias.go

```
StoreKey   = types.StoreKey
```

Recommendation

We recommend client to remove redundant variables.

Alleviation

`[CertiK]` : Supernova team heeded the advice and resolved this finding in commit 0e2e03ca45cf46132c0e6d01768e2936394d3953.

932-03 | REDUNDANT ALIAS

Category	Severity	Location	Status
Coding Style	Informational	app/app.go: 17; x/airdrop/keeper/grpc_query.go: 4; x/airdrop/keeper/msg_server.go: 4; x/gal/types/msgs.go: 5; x/icacontrol/keeper/grpc_query.go: 4; x/icacontrol/keeper/ibc_handler.go: 6; x/mint/types/params.go: 8	Resolved

Description

In the linked positions, the alias are same as module name.

/app/app.go

```
17      gal "github.com/Carina-labs/nova/x/gal"
```

/x/airdrop/keeper/msg_server.go

```
4   context "context"
```

/x/airdrop/keeper/grpc_query.go

```
4   context "context"
```

/x/gal/types/msgs.go

```
5   time "time"
```

/x/icacontrol/keeper/grpc_query.go

```
4   context "context"
```

/x/icacontrol/keeper/ibc_handler.go

```
6   proto "github.com/gogo/protobuf/proto"
```

/x/mint/types/params.go

```
 8      yaml "gopkg.in/yaml.v2"
```

Recommendation

We recommend removing redundant alias.

Alleviation

`[CertiK]` : Supernova team heeded the advice and resolved this finding in commit 1643f190143fbaf9e9068d65efd0cf9527f05b4a.

ANT-01 | UNUSED FUNCTIONS

Category	Severity	Location	Status
Coding Style	Informational	app/ante.go: 31~33	Resolved

Description

The functions in the linked positions are never used.

/app/ante.go

```
func NewMinCommissionDecorator(cdc codec.BinaryCodec) MinCommissionDecorator {
    return MinCommissionDecorator{cdc}
}
```

Recommendation

We recommend removing the unused functions for improving readability.

Alleviation

`[CertiK]` : Supernova team heeded the advice and resolved this finding in commit 5577fa6b7f166595350e29617275a3bf1b4c7242.

GAL-01 | TYPO

Category	Severity	Location	Status
Coding Style	Informational	x/gal/keeper/msg_server.go: 385; x/gal/types/msgs.go: 18	Resolved

Description

`x/gal/keeper/msg_server.go` file

- The error message should be `total withdraw amount: %s".

`x/gal/types/msgs.go` file

- The value of const variable `TypeMsgIcaWithdraw` should be `icaWithdraw` .

Recommendation

We recommend correcting the error message to improve readability.

Alleviation

`[CertiK]` : SuperNove team heeded the advice and resolved the finding in the commit hash `9fdfb6685359c41713c810c4642a051385f3280e` .

IBM-01 | TYPO IN FILE NAME

Category	Severity	Location	Status
Coding Style	Informational	x/icacontrol/ibc_mobule.go: 1	Resolved

Description

"mobule" in file name should be "module".

Recommendation

We recommend renaming the file from "ibc_mobule.go" to "ibc_module.go".

Alleviation

`[CertiK]` : Supernova team heeded the advice and resolved this finding in commit afd74aab8aebab8db262fdea3e70de6678ef5287.

MOU-01 | DUPLICATE CODE

Category	Severity	Location	Status
Coding Style	Informational	app/keepers/modules.go: 71	Resolved

Description

```
68      gal.AppModuleBasic{},
69      icacontrol.AppModuleBasic{},
70      authzmodule.AppModuleBasic{},
71      gal.AppModuleBasic{},
```

The code on line 71 is same as the code on line 68.

Recommendation

We recommend removing the duplicate code.

Alleviation

`[CertiK]` : Supernova team heeded the advice and resolved this finding in commit 75b0621a66a1de3381dc64d4bc55d3a11e69ed42.

MSR-02 | MISSING EMIT EVENTS

Category	Severity	Location	Status
Coding Style	Informational	x/icacontrol/keeper/msg_server.go: 34	Resolved

Description

Functions that update state variables should emit relevant events as notifications.

- RegisterZone()
- DeleteRegisteredZone()
- ChangeRegisteredZone()
- IcaDelegate()
- IcaUndelegate()
- IcaAutoStaking()
- IcaTransfer()
- IcaAuthzGrant()
- IcaAuthzRevoke()
- RegisterControllerAddress()

Recommendation

We recommend adding events for state-changing actions, and emitting them in their relevant functions.

Alleviation

`[CertiK]` : Supernova team heeded the advice and resolved this finding in commit 69edc70d5fd56cc07a4d04b3b04fc9ef7021e009.

MSR-03 | WRONG COMMENTS

Category	Severity	Location	Status
Inconsistency	Informational	x/icacontrol/keeper/msg_server.go: 98	Resolved

Description

The linked comment is incorrect, since it is for the function `ChangeRegisteredZone()`.

Recommendation

We recommend correcting the comment to improve the code readability.

Alleviation

`[CertiK]` : Supernova team heeded the advice and resolved this finding in commit 5d13392f97a2b6af091a33c5a08e6985d32fa18a.

MSR-04 DISCUSSION ON MESSAGE `MsgChangeRegisteredZone` IN MODULE `icacontrol`

Category	Severity	Location	Status
Volatile Code	Informational	x/icacontrol/keeper/msg_server.go: 99	Resolved

Description

The function `ChangeRegisteredZone()` is used to modify any data of the zone, including `zoneId`. We are not sure if it is appropriate to modify the zone id.

How is the team thinking about this?

Recommendation

We recommend reviewing the logic to ensure it meets the design intent.

Alleviation

`[Supernova]`: Human error can occur because all zone information is entered directly by people. Therefore, if the zone Id is entered incorrectly, it is implemented so that it can be corrected.

TXL-01 | UNUSED INPUT ARGUMENTS

Category	Severity	Location	Status
Coding Style	Informational	x/gal/client/cli/tx.go: 23	Resolved

Description

```
16 func txDepositCmd() *cobra.Command {
17   cmd := &cobra.Command{
18     Use:   "deposit [zone-id] [depositor] [claimer] [amount]",
19     Short: "Deposit wrapped token to nova",
20     Long: `Deposit wrapped token to nova.
21 Note, the '--from' flag is ignored as it is implied from [from_key_or_address].
22 When using '--dry-run' a key name cannot be used, only a bech32 address.`,
23     Args: cobra.ExactArgs(4),
```

The `txDepositCmd()` method has four parameters, `zoneId`, `depositor`, `claimer` and `amount`. But the second parameter is never used in this method, which will use `clientCtx.GetFromAddress()` as the address of the `depositor`. This may confuse the caller when setting the input parameters.

Recommendation

We understand this is not an issue, but we recommend removing this unused argument to improve code readability.

Alleviation

`[CertiK]`: Supernova team heeded the advice and resolved this finding in commit 1fe1391998c7cf0d54d8e1a5d336741332fdee0f.

OPTIMIZATIONS | SUPERNOVA

ID	Title	Category	Severity	Status
X93-04	Improper Validation Sequence	Gas Optimization	Optimization	Resolved

X93-04 | IMPROPER VALIDATION SEQUENCE

Category	Severity	Location	Status
Gas Optimization	Optimization	x/gal/keeper/msg_server.go: 102~109; x/icacontrol/keeper/msg_server.go: 149	Resolved

Description

```
102  if !m.keeper.icaControlKeeper.IsValidControllerAddr(ctx, delegate.ZoneId,
delegate.ControllerAddress) {
103    return nil, sdkerrors.Wrap(sdkerrors.ErrInvalidAddress,
delegate.ControllerAddress)
104  }
105
106  zoneInfo, ok := m.keeper.icaControlKeeper.GetRegisteredZone(ctx,
delegate.ZoneId)
107  if !ok {
108    return nil, types.ErrNotFoundZoneInfo
109  }
```

According to the above statement, `delegate.ControllerAddress` will be validated based on `delegate.ZoneId`. After that, the L106 code will validate `delegate.ZoneId`. Since `delegate.ControllerAddress` is based on `delegate.ZoneId`, we recommend validating `delegate.ZoneId` first.

There are similar cases in the functions `Delegate()`, `Undelegate()`, `IcaWithdraw()`, and `ClaimSnAsset()`, and functions in `/x/icacontrol/keeper/msg_server.go` file.

Recommendation

We recommend validating `delegate.ZoneId` first.

Alleviation

`[CertiK]`: Supernova team heeded the advice and resolved the finding in the commit hash `796d414600a2b6da654bdae10236f7d8e8b6e587`.

APPENDIX | SUPERNOVA

Finding Categories

Categories	Description
Gas Optimization	Gas Optimization findings do not affect the functionality of the code but generate different, more optimal EVM opcodes resulting in a reduction on the total gas cost of a transaction.
Logical Issue	Logical Issue findings detail a fault in the logic of the linked code, such as an incorrect notion on how block.timestamp works.
Volatile Code	Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that may result in a vulnerability.
Language Specific	Language Specific findings are issues that would only arise within Solidity, i.e. incorrect usage of private or delete.
Coding Style	Coding Style findings usually do not affect the generated byte-code but rather comment on how to make the codebase more legible and, as a result, easily maintainable.
Inconsistency	Inconsistency findings refer to functions that should seemingly behave similarly yet contain different code, such as a constructor assignment imposing different require statements on the input variables than a setter function.

Checksum Calculation Method

The "Checksum" field in the "Audit Scope" section is calculated as the SHA-256 (Secure Hash Algorithm 2 with digest size of 256 bits) digest of the content of each file hosted in the listed source repository under the specified commit.

The result is hexadecimal encoded and is the same as the output of the Linux "sha256sum" command against the target file.

DISCLAIMER | CERTIK

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to you ("Customer" or the "Company") in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes, nor may copies be delivered to any other person other than the Company, without CertiK's prior written consent in each instance.

This report is not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts CertiK to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. CertiK's position is that each company and individual are responsible for their own due diligence and continuous security. CertiK's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

The assessment services provided by CertiK is subject to dependencies and under continuing development. You agree that your access and/or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon, multiple layers of third-parties.

ALL SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF ARE PROVIDED "AS IS" AND "AS AVAILABLE" AND WITH ALL FAULTS AND DEFECTS WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, CERTIK HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, CERTIK SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AND ALL WARRANTIES ARISING FROM COURSE OF DEALING, USAGE, OR TRADE PRACTICE. WITHOUT LIMITING THE FOREGOING, CERTIK MAKES NO WARRANTY OF ANY KIND THAT THE SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF, WILL MEET CUSTOMER'S OR ANY OTHER PERSON'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULT, BE COMPATIBLE OR WORK WITH ANY SOFTWARE, SYSTEM, OR OTHER SERVICES, OR BE SECURE, ACCURATE, COMPLETE, FREE OF HARMFUL CODE, OR ERROR-FREE. WITHOUT LIMITATION TO THE

FOREGOING, CERTIK PROVIDES NO WARRANTY OR UNDERTAKING, AND MAKES NO REPRESENTATION OF ANY KIND THAT THE SERVICE WILL MEET CUSTOMER'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULTS, BE COMPATIBLE OR WORK WITH ANY OTHER SOFTWARE, APPLICATIONS, SYSTEMS OR SERVICES, OPERATE WITHOUT INTERRUPTION, MEET ANY PERFORMANCE OR RELIABILITY STANDARDS OR BE ERROR FREE OR THAT ANY ERRORS OR DEFECTS CAN OR WILL BE CORRECTED.

WITHOUT LIMITING THE FOREGOING, NEITHER CERTIK NOR ANY OF CERTIK'S AGENTS MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND, EXPRESS OR IMPLIED AS TO THE ACCURACY, RELIABILITY, OR CURRENCY OF ANY INFORMATION OR CONTENT PROVIDED THROUGH THE SERVICE. CERTIK WILL ASSUME NO LIABILITY OR RESPONSIBILITY FOR (I) ANY ERRORS, MISTAKES, OR INACCURACIES OF CONTENT AND MATERIALS OR FOR ANY LOSS OR DAMAGE OF ANY KIND INCURRED AS A RESULT OF THE USE OF ANY CONTENT, OR (II) ANY PERSONAL INJURY OR PROPERTY DAMAGE, OF ANY NATURE WHATSOEVER, RESULTING FROM CUSTOMER'S ACCESS TO OR USE OF THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS.

ALL THIRD-PARTY MATERIALS ARE PROVIDED "AS IS" AND ANY REPRESENTATION OR WARRANTY OF OR CONCERNING ANY THIRD-PARTY MATERIALS IS STRICTLY BETWEEN CUSTOMER AND THE THIRD-PARTY OWNER OR DISTRIBUTOR OF THE THIRD-PARTY MATERIALS.

THE SERVICES, ASSESSMENT REPORT, AND ANY OTHER MATERIALS HEREUNDER ARE SOLELY PROVIDED TO CUSTOMER AND MAY NOT BE RELIED ON BY ANY OTHER PERSON OR FOR ANY PURPOSE NOT SPECIFICALLY IDENTIFIED IN THIS AGREEMENT, NOR MAY COPIES BE DELIVERED TO, ANY OTHER PERSON WITHOUT CERTIK'S PRIOR WRITTEN CONSENT IN EACH INSTANCE.

NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS.

THE REPRESENTATIONS AND WARRANTIES OF CERTIK CONTAINED IN THIS AGREEMENT ARE SOLELY FOR THE BENEFIT OF CUSTOMER. ACCORDINGLY, NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH REPRESENTATIONS AND WARRANTIES AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH REPRESENTATIONS OR WARRANTIES OR ANY MATTER SUBJECT TO OR RESULTING IN INDEMNIFICATION UNDER THIS AGREEMENT OR OTHERWISE.

FOR AVOIDANCE OF DOUBT, THE SERVICES, INCLUDING ANY ASSOCIATED ASSESSMENT REPORTS OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

CertiK | Securing the Web3 World

Founded in 2017 by leading academics in the field of Computer Science from both Yale and Columbia University, CertiK is a leading blockchain security company that serves to verify the security and correctness of smart contracts and blockchain-based protocols. Through the utilization of our world-class technical expertise, alongside our proprietary, innovative tech, we're able to support the success of our clients with best-in-class security, all whilst realizing our overarching vision; provable trust for all throughout all facets of blockchain.