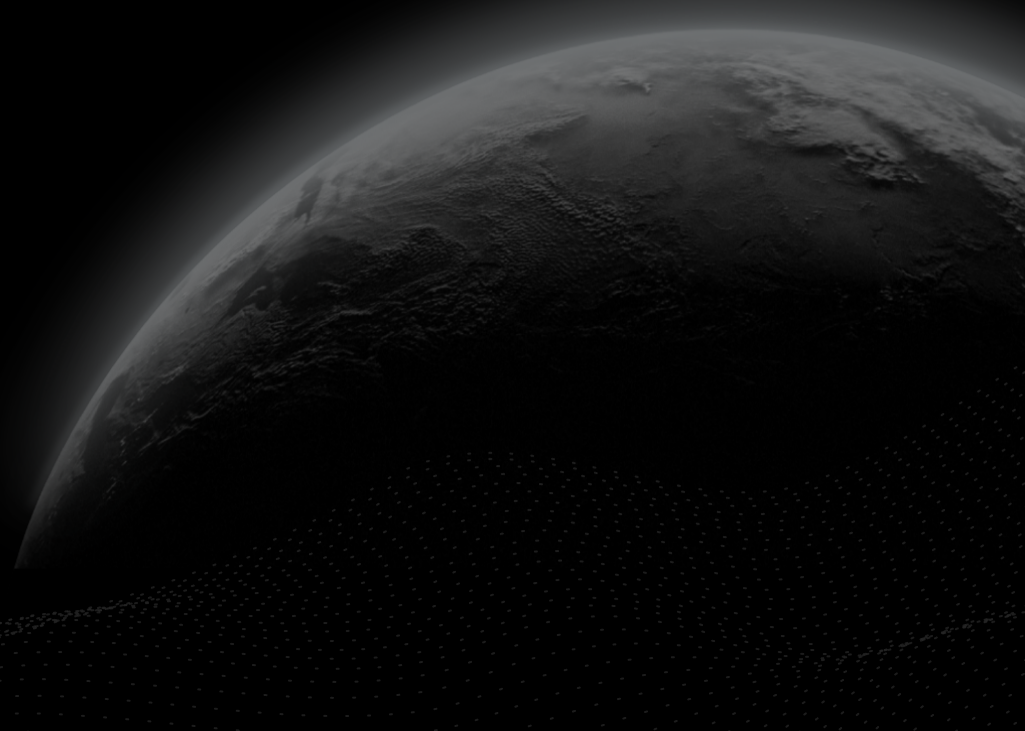
CERTIK

Security Assessment

Stargaze

CertiK Verified on Apr 17th, 2023



CertiK Verified on Apr 17th, 2023

Stargaze

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

Executive Summary

TYPES	ECOSYSTEM	METHODS
DeFi	Cosmos (ATOM)	Manual Review, Static Analysis

LANGUAGE	TIMELINE	KEY COMPONENTS
Golang	Delivered on 04/17/2023	N/A

CODEBASE

https://github.com/public-awesome/stargaze/tree/b31a13e81a927d11449d2e9a176cd9d5a914039d

...View All

COMMITS

b31a13e81a927d11449d2e9a176cd9d5a914039d

...View All

Vulnerability Summary

4	4	0	0	0	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.
0	Major		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.
1	Medium	1 Resolved	Medium risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform.
2	Minor	2 Resolved	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.
1	Informational	1 Resolved	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 | STARGAZE

CODEBASE | STARGAZE

Repository

https://github.com/public-awesome/stargaze/tree/b31a13e81a927d11449d2e9a176cd9d5a914039d

Commit

b31a13e81a927d11449d2e9a176cd9d5a914039d

AUDIT SCOPE | STARGAZE

21 files audited ● 4 files with Resolved findings ● 17 files without findings

ID	File	SHA256 Checksum
● ABC	abci.go	05971b34a33574472c6087a407355fcadb196cffada84297029ce9478d142c40
● TXC	client/cli/tx.go	8b871136b75b6808ac99fbd239194cad73ffd2e2f046f4daa0442d4aa868776b
● TXR	client/rest/tx.go	f647aee7ead90f36d7bd942bcc744628d2a652ee1963bf9a438b175da5a99bd3
● PRI	keeper/privileged.go	8b380afe5084e8f661d8c240d9aca0bcba6b82b193eb39b49a49e4c8f3d53cbc
● GEN	genesis.go	bb532afdd04d17568e859704771769d02ceb7f9d6ff8c8c10c53e3568d744c2e
● MOD	module.go	55e9fd26c312f63440d351710f5ed00719c3da2fea5f023d2dc148a38e152958
● QUE	client/cli/query.go	9438df65640d1fd00062b652427b5ec510bef7c3ba76bc832aa38ff139e239b9
● PRO	client/proposal_handler.go	c97ce47c3b8812bc94c0ec5b1ccea99c9707f0569e818c4819f6d08a9c6f4ae3
● CAL	contract/callback_msgs.go	cd604479965b6d88f05365b5e2ad6028df03588644eeef1f1d9f19aa772cf144
● GRP	keeper/grpc_query.go	a683dfb70d735cd8f479b37939282b14fb8684188c93aa68f97041c1193dce6b
● KEE	keeper/keeper.go	239275fa53e19e221c7d1663bb4dd8066ad51ab8fd68cfdb63862b3cbade35f9
● MSG	keeper/msg_server.go	b82487cf0fef5247a725c6ccd06a951e0952003ac69e84751687f72771e6147e
● COD	types/codec.go	0278f30ee7a98360840ebaeec8039c0165d342ce2b0ba8006957aa49163308d5
● ERR	types/errors.go	41a52535a620eee923ff3fbda2a69584626539b92a2ced5e216da90169a14995

ID	File	SHA256 Checksum
EVE	types/events.go	d25cafc3290a294a8d95d0b116b4ceccf04cf9725370cdbacbfac13d583a0d4a
EXP	types/expected_keepers.go	a42fa476426e3f35a4b88547772c8e2f2a6f6418220a84b74ed4389aee573b9b
GEE	types/genesis.go	fb458e157b939c63b1e59495d077ce1839878656250d4ed5d2494f5e819809af
KEY	types/keys.go	bf42b4a9a82c674f70c6eebe45bcbbd5448e95dd5069c1dba57a480dd00f42b0
PAR	types/params.go	4bc10cd1ab904b638aeaffa3b92a56cb75cfb2a8aa06006ba7d3983fb3b0b89a
PRP	types/proposal.go	aecf705da6520d43c3718213e854c0f3a8caa5375ad269d1d43d01a015a04c46
TYP	types/types.go	7c347886dbeed39a02f9f23d860ffb46fa1da70151c2268a6289325c55acf415

APPROACH & METHODS | STARGAZE

This report has been prepared for Stargaze to discover issues and vulnerabilities in the source code of the Stargaze 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.

FINDINGS | STARGAZE

4	0	0	1	2	1
Total Findings	Critical	Major	Medium	Minor	Informational

This report has been prepared to discover issues and vulnerabilities for Stargaze. Through this audit, we have uncovered 4 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
TXC-01	Commands Not Added	Control Flow	Medium	Resolved
ABC-01	Incorrect Key Used	Logical Issue	Minor	Resolved
TXR-01	A Mismatched Request Structure Is Used	Inconsistency	Minor	Resolved
TXC-02	Failure On Tx Cmd	Volatile Code	Informational	Resolved

TXC-01 | COMMANDS NOT ADDED

Category	Severity	Location	Status
Control Flow	Medium	client/cli/tx.go: 35, 76	Resolved

Description

The `ProposalUnsetPrivilegeContractCmd` and `ProposalSetPrivilegeContractCmd` commands are currently not properly integrated into the parent command, and as a result, they may not function correctly as tx commands.

Recommendation

To ensure proper functionality of the `ProposalUnsetPrivilegeContractCmd` and `ProposalSetPrivilegeContractCmd commands` as tx commands, it is recommended to add them to the `GetTxCmd()` function. Please see the code snippet below for reference, and be sure to thoroughly test the implementation:

```
func GetTxCmd() *cobra.Command {

    ......

    // this line is used by starport scaffolding # 1
    cmd.AddCommand(ProposalSetPrivilegeContractCmd())
    cmd.AddCommand(ProposalUnsetPrivilegeContractCmd())

    return cmd
}
```

Alleviation

`[Stargaze]`: Governance tx commands are added under `starsd tx gov submit-proposal` command wired in app.go line 174.

ABC-01 | INCORRECT KEY USED

Category	Severity	Location	Status
Logical Issue	Minor	abci.go: 20	Resolved

Description

In Cosmos, the `telemetry.ModuleMeasureSince` function is used to measure the elapsed time since a specific event occurred in a module. When using this function in the `BeginBlocker()` function of a Cosmos SDK module, it is important to use the correct metric key for the event being measured.

```
func BeginBlocker(ctx sdk.Context, k keeper.Keeper, w types.WasmKeeper) {
    defer telemetry.ModuleMeasureSince(types.ModuleName, time.Now(),
telemetry.MetricKeyEndBlocker)
    sudoMsg := contract.SudoMsg{BeginBlock: &struct{}{}}
    k.IteratePrivileged(ctx, abciContractCallback(ctx, w, sudoMsg))
}
```

The correct metric key for measuring the elapsed time of the `BeginBlocker()` function is `telemetry.MetricKeyBeginBlocker`.

Recommendation

It's recommended to change the second argument of `telemetry.ModuleMeasureSince` to `telemetry.MetricKeyBeginBlocker`.

Alleviation

`[CertiK]`: The team heeded the advice and resolved this issue in commit `38430119e1e62d1582d5be97fecc873bb26524bd`.

TXR-01 | A MISMATCHED REQUEST STRUCTURE IS USED

Category	Severity	Location	Status
Inconsistency	Minor	client/rest/tx.go: 32	Resolved

Description

The type of `req` should be `UnsetPrivilegeProposalJSONReq`, the below code segment uses a mismatched request.

```
 28 func ProposalDemotePrivilegeContractHandler(cliCtx client.Context)
govrest.ProposalRESTHandler {
 29     return govrest.ProposalRESTHandler{
 30         SubRoute: "wasm_demote_privilege",
 31         Handler: func(w http.ResponseWriter, r *http.Request) {
 32             var req SetPrivilegeProposalJSONReq
 33             if !rest.ReadRESTReq(w, r, cliCtx.LegacyAmino, &req) {
 34                 return
 35             }
 36             toStdTxResponse(cliCtx, w, req)
 37         },
 38     }
 39 }
```

Recommendation

While the code logic appears to be working correctly at present, it is strongly recommended that the client double-checks it to ensure there are no issues in the future.

Alleviation

`[CertiK]`: The team heeded the advice and resolved this issue in commit `38430119e1e62d1582d5be97fecc873bb26524bd`.

TXC-02 | FAILURE ON TX CMD

Category	Severity	Location	Status
Volatile Code	Informational	client/cli/tx.go: 58	Resolved

Description

The `readme` for the current module provides some helpful `tx` command examples, but it has been found that some of them may not function as expected. For instance, the `promote-contract` command example is shown below:

```
starsd tx cron promote-contract {contractAddr} --title {proposalTitle} --deposit
{depositAmount}
starsd tx cron promote-contract
stars19jq6mj84cnt9p7sagjxqf8hxtczwc8wlpuwe4sh62w45aheseues57n420 --title "Promote
Contract Proposal" --deposit 1000ustars
```

When executing this command directly, it generates an error message indicating an invalid address. Upon further investigation, it has been determined that this error is due to the lack of a proposer, i.e., it cannot obtain the `from` address of the `clientCtx`. Additionally, the `from` flag for this command is not recognized.

Recommendation

We recommend reviewing the logic again.

Alleviation

`[Stargaze]`: The correct usage of this command should be through the gov module, the commands are `starsd tx gov submit-proposal promote-contract` and `starsd tx gov submit-proposal demote-contract`.

OPTIMIZATIONS | STARGAZE

ID	Title	Category	Severity	Status
PRI-01	Optimization On Function `SetPrivileged()`	Gas Optimization	Optimization	Resolved

PRI-01 | OPTIMIZATION ON FUNCTION `SetPrivileged()`

Category	Severity	Location	Status
Gas Optimization	Optimization	keeper/privileged.go: 12	Resolved

Description

The `SetPrivileged()` function is used to add a contract to the list of privileged contracts or update it if it is already in the list. However, if the contract is already privileged, there is no need to update the store and this can be an unnecessary gas cost.

Recommendation

To optimize the `SetPrivileged()` function, it can first check if the given contract is already privileged, and if so, do nothing. Here's an example of how you can modify the function to include this optimization:

```
func (k Keeper) SetPrivileged(ctx sdk.Context, contractAddr sdk.AccAddress) error {
    if k.wasmKeeper.HasContractInfo(ctx, contractAddr) {
        if !k.IsPrivileged(ctx, contractAddr) {
          // coding here ...

        else {
          return nil
        }

    } else {
        return types.ErrContractDoesNotExist
    }
    return nil
}
```

Alleviation

`[CertiK]`: The team heeded the advice and resolved this issue in commit `38430119e1e62d1582d5be97fecc873bb26524bd`.

APPENDIX | STARGAZE

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.
Control Flow	Control Flow findings concern the access control imposed on functions, such as owner-only functions being invoke-able by anyone under certain circumstances.
Volatile Code	Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that may result in a vulnerability.
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.