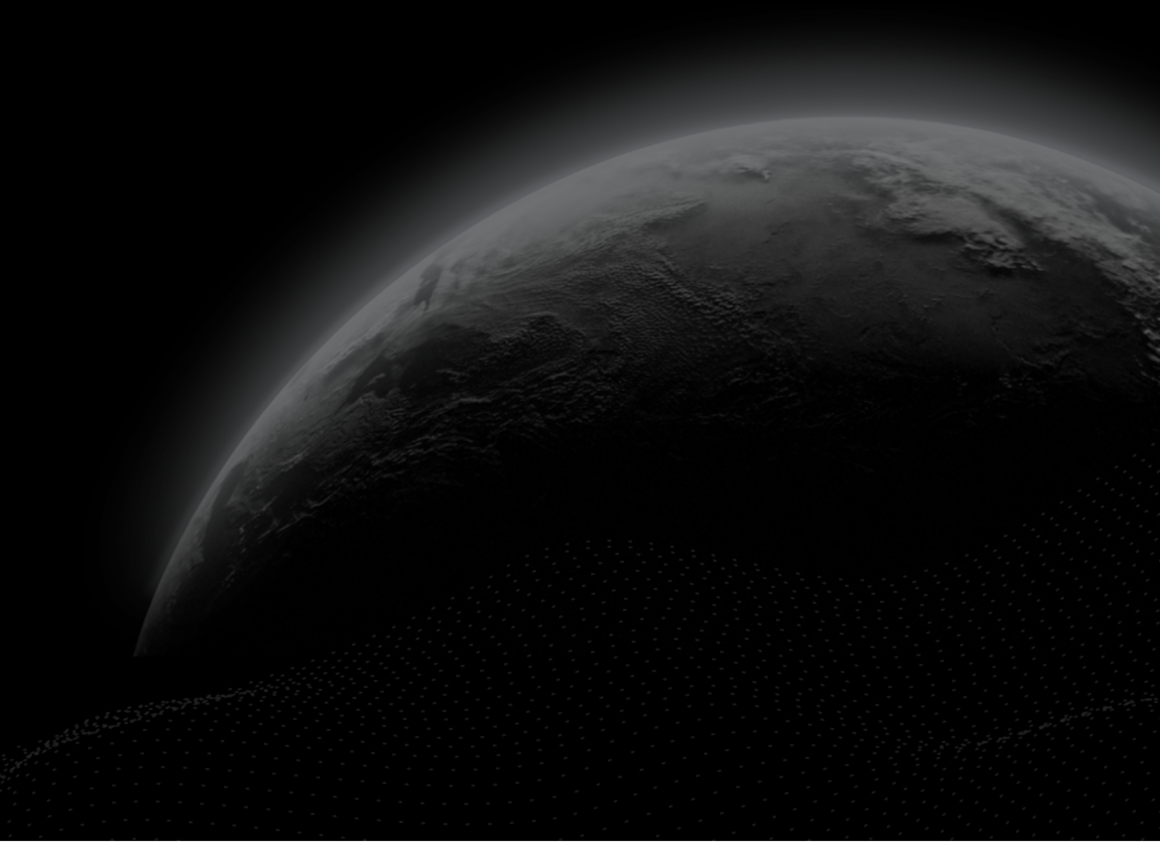
Security Assessment

# NxvVault Wallet-audit

CertiK Assessed on Jan 31st, 2024



CertiK Assessed on Jan 31st, 2024

# NxvVault Wallet-audit

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

## Executive Summary

| TYPES | ECOSYSTEM | METHODS |
|---|---|---|
| DeFi | Binance Smart Chain (BSC) \| Ethereum (ETH) | Manual Review, Static Analysis |

| LANGUAGE | TIMELINE | KEY COMPONENTS |
|---|---|---|
| Solidity | Delivered on 01/31/2024 | N/A |

CODEBASE

https://github.com/nexvault/safe-contracts/tree/b91255866d904720503bed85d521803440b5e284

View All in Codebase Page

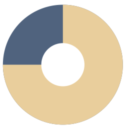
COMMITS

b91255866d904720503bed85d521803440b5e284

bbae913b03254eaddc596ebf02e673eeeef99d75

View All in Codebase Page

## Vulnerability Summary



| 4 | 1 | 0 | 0 | 3 | 0 |
|---|---|---|---|---|---|
| Total Findings | Resolved | Mitigated | Partially Resolved | Acknowledged | Declined |

| | | | | |
|---|---|---|---|---|
| 🟥 | 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. |
| 🟨 | 0 | Medium | | Medium risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform. |
| 🟨 | 3 | Minor | 3 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. |
| 🟦 | 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 | NXVVAULT WALLET-AUDIT

# CODEBASE | NXVVAULT WALLET-AUDIT

## Repository

https://github.com/nexvault/safe-contracts/tree/b91255866d904720503bed85d521803440b5e284

## Commit

b91255866d904720503bed85d521803440b5e284

bbae913b03254eaddc596ebf02e673eeeef99d75

# AUDIT SCOPE | NXVVAULT WALLET-AUDIT

5 files audited ● 5 files without findings

| ID | Repo | File | SHA256 Checksum |
|---|---|---|---|
| ● NXC | nexvault/nxv-safe-wallet | contracts/NXV.sol | 896006fb1159e37d8297e75e499210f589601c77b93c0aee6358d85224a3048d |
| ● NVM | nexvault/nxv-safe-wallet | contracts/libraries/NXVMigration.sol | 99f863b2e30613c126d1a07d33ce5c81918bcdc08e80d380f83683ae80e686dd |
| ● NVS | nexvault/nxv-safe-wallet | contracts/libraries/NXVStorage.sol | 86d2e0125ea1565d5de525821897b93b8bcc137c6d220279cdfe01249bb67538 |
| ● SIG | nexvault/nxv-safe-wallet | contracts/libraries/SignMessageLib.sol | da74c979662164d26f71ca2ae4733c54a2386d725d88a35d2e8ef8b092252350 |
| ● NXF | nexvault/nxv-safe-wallet | contracts/proxies/NXVProxyFactory.sol | 547f3106e2db16cab9968be31457e19fd5d404ed45cb05ca4f01553e044bb4f0 |

# APPROACH & METHODS | NXVVAULT WALLET-AUDIT

This report has been prepared for NxvVault to discover issues and vulnerabilities in the source code of the NxvVault Wallet-audit 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:

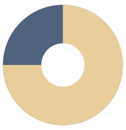
- Testing the smart contracts against both common and uncommon attack vectors;
- Enhance general coding practices for better structures of source codes;
- Add enough unit tests to cover the possible use cases;
- Provide more comments per each function for readability, especially contracts that are verified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

# REVIEW NOTES | NXVVAULT WALLET-AUDIT

The "safe-contracts" for "NexVault" are forked from the "https://github.com/safe-global/safe-smart-account" project. The auditor cross-examined the codebase to identify the files with code modifications other than variable name and comment changes, which are as follows. The following code was audited in the review:

- contracts/
    - NXV.sol
- contracts/libraries
    - NXVMigration.sol
    - NXVStorage.sol
    - SignMessageLib.sol
- contracts/libraries
    - NXVProxyFactory.sol

# FINDINGS | NXVVAULT WALLET-AUDIT



| Total Findings | Critical | Major | Medium | Minor | Informational |
|---|---|---|---|---|---|
| 4 | 0 | 0 | 0 | 3 | 1 |

This report has been prepared to discover issues and vulnerabilities for NxvVault Wallet-audit. 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 |
|---|---|---|---|---|
| CON-02 | Lack Of Access Control | Access Control | Minor | ● Acknowledged |
| GLOBAL-01 | Out Of Scope Dependency | Access Control | Minor | ● Acknowledged |
| NXV-02 | Signature Malleability Of `ecrecover` | Volatile Code | Minor | ● Acknowledged |
| NXP-01 | Lack Of Comments And Mismatch Between Comments And Implementation | Coding Style | Informational | ● Resolved |

# CON-02 | LACK OF ACCESS CONTROL

| Category | Severity | Location | Status |
|---|---|---|---|
| Access Control | ● Minor | NXV.sol (b912558): 88; FallbackManager.sol (b912558): 50 | ● Acknowledged |

## Description

The function `setup()` can be called by anyone as it has no access restriction. This enables anyone to call this and set an initial storage of the NXV contract. If a proxy was created without setting up, anyone can call setup and claim the proxy.

## Recommendation

We recommend ensuring that the function setup() is called when the proxy is created.

## Alleviation

The team acknowledged the finding and decided not to change the current codebase.

**[NxvVault Team, 01/29/2024]**: In combination with the platform's business scenarios, all Proxy wallet contracts will be created through a factory contract, and will be initialized during the creation process by calling the setup() function of the Proxy wallet contract. The platform will not process any Proxy wallet contracts that are not created normally.

# GLOBAL-01 | OUT OF SCOPE DEPENDENCY

| Category | Severity | Location | Status |
|---|---|---|---|
| Access Control | ● Minor | | ● Acknowledged |

## Description

The multi-sig safe contract serves as a middleware proxy, calling other user-supplied target contract addresses with user-supplied input data. It allows performing 'call' or 'delegatecall' to the target contract. Calling an arbitrary contract with a 'delegatecall' is dangerous, as the `delegatecall` function enables a contract to call another contract's function within the context of the caller's state. If the callee is untrusted and malicious, it can result in the caller contract modifying its storage in an unexpected way. If a self-destructor is used in the target contract, it could affect the multi-sig contract as well. The audit considers the contracts called through the multi-sig safe contract as out of scope, and users who use the multi-sig contract should ensure that their calls are to trusted contracts.

## Recommendation

It's recommended that the project remind users to ensure they only interact with trusted contract addresses. Also, remind users of the implications of using a 'delegatecall', and to avoid 'delegatecall' if not necessary.

## Alleviation

The team acknowledged the finding and decided not to change the current codebase.

**[NxvVault Team, 01/29/2024]**: For the current contract architecture, the delegatecall method is essential, particularly in scenarios involving logic contract upgrades and flexible external library calls. For the Nexvault platform, all transaction data created through the platform undergoes strict parameter parsing and risk assessment.

# NXV-02 | SIGNATURE MALLEABILITY OF `ecrecover`

| Category | Severity | Location | Status |
|---|---|---|---|
| Volatile Code | ● Minor | NXV.sol (b912558): 132, 155, 178 | ● Acknowledged |

## Description

The `ecrecover()` function is subject to signature malleability. The signature malleability is possible within the Elliptic Curve cryptographic system. An Elliptic Curve is symmetric on the `X` -axis, meaning two points can exist with the same `X` value. In the `r` , `s` and `v` representation this permits us to carefully adjust `s` to produce a second valid signature for the same `r` , thus breaking the assumption that a signature cannot be replayed in what is known as a replay-attack.

```
132        checkSignatures(txHash, "", signatures);
```

- `checkSignatures` called.

```
155        checkNSignatures(txHash, data, signatures, _threshold);
```

- `checkNSignatures` called.

```
178            currentOwner = ecrecover(txHash, v, r, s);
```

- `ecrecover` called without proper checks.

## Recommendation

We recommend adding the following checks or to consider the example in `ECDSA.sol` from the OpenZeppelin library.

```
require(uint256(s) <= 0x7FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF5D576E7357A4501DDFE92F46681B20A0, "ECDSA: invalid signature 's' value");
require(uint8(v) == 27 || uint8(v) == 28, "ECDSA: invalid signature 'v' value");
```

## Alleviation

The team acknowledged the finding and decided not to change the current codebase.

**[NxvVault Team, 01/29/2024]**: Because we have integrated third-party wallets to sign transaction information using EIP-712, we cannot guarantee that the handling of signatures by third-party wallets strictly adheres to the EIP-2 standard. For the parties we integrate with, we do not wish to alter the signature results of third-party wallets by reversing the changes to the 's' value.

# NXP-01 LACK OF COMMENTS AND MISMATCH BETWEEN COMMENTS AND IMPLEMENTATION

| Category | Severity | Location | Status |
|---|---|---|---|
| Coding Style | ● Informational | NXVProxyFactory.sol (b912558): 108~138 | ● Resolved |

## Description

Some of the comments in the codebase do not match the implementation. This is likely because the code was forked from another project, and the comments have not been adjusted to match the current implementation. For example, the "Guard" and "Modules" features have been removed, but the comments remain in the code.

```
 *      - Guard: Guard is a contract that can execute pre- and post-transaction
checks. Managed in `GuardManager`.
 *      - Modules: Modules are contracts that can be used to extend the write
functionality of a Safe. Managed in `ModuleManager`.
```

For the two newly added functions, "createProxyWithTransaction" and "calculateNXVAddress" in the "NXVProxyFactory.sol", the comments are missing compared to other functions in the same file.

## Recommendation

It is recommended that the team modify the comments in the code to ensure they match the current implementation and also insert comments for newly added functions.

## Alleviation

**[NxvVault Team, 01/29/2024]**: The team heeded the advice and resolved the issue in commit 66e99ba15d7b0129d9226110dcf03613f43be4c5.

# OPTIMIZATIONS | NXVVAULT WALLET-AUDIT

| ID | Title | Category | Severity | Status |
|---|---|---|---|---|
| CON-03 | Inefficient Memory Parameter | Inconsistency | Optimization | ● Acknowledged |

# CON-03 | INEFFICIENT MEMORY PARAMETER

| Category | Severity | Location | Status |
|---|---|---|---|
| Inconsistency | ● Optimization | NXV.sol (b912558): 120; NXVProxyFactory.sol (b912558): 80, 99, 110, 117, 125 | ● Acknowledged |

## Description

One or more parameters with `memory` data location are never modified in their functions and those functions are never called internally within the contract. Thus, their data location can be changed to `calldata` to avoid the gas consumption copying from calldata to memory.

```
    function execTransaction(
```

`execTransaction` has memory location parameters: `signatures` .

```
    function createChainSpecificProxyWithNonce(
```

`createChainSpecificProxyWithNonce` has memory location parameters: `initializer` .

```
    function createProxyWithCallback(
```

`createProxyWithCallback` has memory location parameters: `initializer` .

```
    function createProxyWithTransaction(
```

`createProxyWithTransaction` has memory location parameters: `initializer` , `signatures` .

```
    function calculateNXVAddress(
```

`calculateNXVAddress` has memory location parameters: `initializer` .

## Recommendation

We recommend changing the parameter's data location to `calldata` to save gas.

- For Solidity versions prior to 0.6.9, since public functions are not allowed to have calldata parameters, the function visibility also needs to be changed to `external` .

- For Solidity versions prior to 0.5.0, since parameter data location is implicit, changing the function visibility to `external` will change the parameter's data location to calldata as well.

## Alleviation

The team acknowledged the finding and decided not to change the current codebase.

**[NxvVault Team, 01/29/2024]**: After modifying the test according to the suggested method, it was found that the Gas consumption did not significantly decrease.

# APPENDIX | NXVVAULT WALLET-AUDIT

## Finding Categories

| Categories | Description |
|---|---|
| Coding Style | Coding Style findings may not affect code behavior, but indicate areas where coding practices can be improved to make the code more understandable and maintainable. |
| Access Control | Access Control findings are about security vulnerabilities that make protected assets unsafe. |
| Inconsistency | Inconsistency findings refer to different parts of code that are not consistent or code that does not behave according to its specification. |
| Volatile Code | Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases and may result in vulnerabilities. |

## 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.