

Smart Contract Audit Report

T2T2



Sept. 6th, 2023

Copyright

All rights reserved by SharkTeam. Unless otherwise specified, the copyright or other related rights of any text description, document format, illustration, photo, method, process, etc. in this document belong to SharkTeam. Without the written consent of SharkTeam, no one is allowed to copy, extract, back up, modify, disseminate, translate into other languages or use all or part of this manual for commercial purposes in any way or form.

1. Overview

SharkTeam recently received the requirements for T2T2 smart contracts audit. In this audit, the SharkTeam security experts communicate with T2T2 team to conduct smart contract security audit under controllable operation, so as to avoid any risk in the audit process as far as possible.

Project Overview :

Project Name	T2T2
Description	NFT
Language	Solidity
Codebase	Private Pro
MD5	8bd3ab28e34e8f4367c64b3a989d68ba

Audit method :

Firstly, through static analysis, dynamic analysis and other analysis technologies, the smart contracts in the project were automatically scanned and manually reviewed; after that, the SharkTeam security experts conducted a detailed manual audit of the smart contracts line-by-line. From the four dimensions of high-level language, virtual machine, blockchain, and business logic, nearly 200 audit items of smart contracts have been comprehensively audited.

Audit Items:

For the Solidity smart contract, SharkTeam's audit items cover four layers: language, virtual machine, blockchain, and business, and four levels of high risk, medium risk, low risk, and information. Some of the common high-risk items (not all vulnerabilities) are as follows:

- Function Visibility and Access Control
- Reentrancy Vulnerability
- Contract and Storage Initialization
- Token Minting and Burning
- Centralization of Power
- Function Logic Vulnerability
- Flashloan and Price Manipulation
- DAO Proposal Attack
- Contract Upgrade Issues
- Randomness and Revert Attack
- Insufficient Randomness
- Integer Overflow/Underflow
- Divide/Multiply and Integer Precision
- Unchecked/Unused Return Values
- Blockchain Timestamp Dependency
- Transaction Order Dependency and Front Running
- Denial of Service (DoS)

Audit Scope :

Contract Files	MD5
T2MultiverseSoulShares/ERC165.sol	761f329616023b99424de04cf5791148
T2MultiverseSoulShares/T2MultiverseSoulShares.sol	e9cd5bbe9d93a452d48ff2e2b231c4b7
T2MultiverseSoulShares/common/IVerifySig.sol	98c0f09427ef6c299cc816c8f5e4c0d1
T2MultiverseSoulShares/utills/Adminable.sol	75741078aee80e7fffd84f51f7e30c1c
T2MultiverseSoulShares/utills/Utilities.sol	1df99f1cfc5cf8ecd376f69d98985e78
ERC6551_TBA/ERC721.sol	650995166348be1ec3184c7925a837fa
ERC6551_TBA/ERC721Enumerable.sol	f359e0559cf3633de398a3cd1ca17dc3
ERC6551_TBA/IERC721.sol	672276d42991c91d7ad1d1f5650136d4
ERC6551_TBA/IERC721Enumerable.sol	7e235d4b86c5433f93fc9b7c1cdaacbc
ERC6551_TBA/IERC721Metadata.sol	0dc3820d7e82cb1a16a5998de45705c2
ERC6551_TBA/IERC721Receiver.sol	3d4292e6bc8ee1bfd0fbb0fee72daa79
ERC6551_TBA/TBA.sol	b1c75a15d0d04cc0cfbb19d1073d1d87
ERC6551_TBA/common/IERC6551Registry.sol	b35a2d363297cfd4c642963773212419
ERC6551_TBA/common/IVerifySig.sol	98c0f09427ef6c299cc816c8f5e4c0d1
ERC6551_TBA/utills/Address.sol	750b244702a16ef3d8e27714bf457403
ERC6551_TBA/utills/Adminable.sol	75741078aee80e7fffd84f51f7e30c1c
ERC6551_TBA/utills/Counters.sol	6c0b46a5400dca01a7d8d15a62e59890
ERC6551_TBA/utills/SafeMath.sol	363045c058e36662bacaa0a7734cd74a
ERC6551_TBA/utills/Utilities.sol	1df99f1cfc5cf8ecd376f69d98985e78

Audit results :

T2T2 smart contract audit results: **Pass**.

SharkTeam

2. Findings

2.1 Summary

Vulnerability list :

ID	Item	Severity	Category	Status
1	Parameter-Chainid-Vulnerability	■ Low	Language	Fixed
2	Invalid-Check	■ Low	Language	Fixed
3	Unreached-Code	■ Info	Language	Fixed
4	Invalid-Library	■ Info	Language	Fixed
5	Centralization-Risk	■ Info	Business	ⓘ Unfixed

2.2 Detailed Results

2.2.1 Centralization-Risk [Info]

Description:

Owner and admin are centralized high-privilege accounts that are closely related to the core business. Only these accounts can update some key state variables. Once the private key is leaked or stolen, the project will suffer a devastating blow.

Recommendation:

Properly keep all private keys of high-privilege accounts secure. And try to avoid the loss, leakage, theft, etc. of any private key as much as possible.

Appendix A: Smart Contract Audit Items

ID	Item	Category	Severity
TVE-001	Different-Pragma-Directives-Are-Used	Language	Info
TVE-002	Incorrect-Versions-Of-Solidity	Language	Info
TVE-003	Solidity-Version-Is-Outdated	Language	Info
TVE-004	Reentrancy-Eth-Vulnerabilities	EVM	High
TVE-005	Reentrancy-No-Eth-Vulnerabilities	EVM	Medium
TVE-006	Reentrancy-Benign-Vulnerabilities	EVM	Low
TVE-007	Reentrancy-Events-Vulnerabilities	EVM	Low
TVE-008	Reentrancy-Unlimited-Gas-Vulnerabilities	EVM	Info
TVE-009	Erc777-Callbacks-And-Reentrancy	Language	High
TVE-010	State-Variable-Shadowing	Language	High
TVE-011	State-Variable-Shadowing-From-Abstract-Contracts	Language	Medium
TVE-012	Builtin-Symbol-Shadowing	Language	Low
TVE-013	Local-Variable-Shadowing	Language	Low
TVE-014	Uninitialized-Local-Variables	Language	Medium
TVE-015	Uninitialized-Storage-Variables	Language	High
TVE-016	Uninitialized-State-Variables	Language	High
TVE-017	Dos-Attack-Call-Failed	Language	Medium

TVE-018	Dos-With-Block-Gas-Limit	EVM	Medium
TVE-019	Unused-State-Variable	Language	Info
TVE-020	Variable-Names-Too-Similar	Language	Info
TVE-021	State-Variables-That-Could-Be-Declared-Constant	Business	Info
TVE-022	Local-Variables-Are-Not-Used	Language	Info
TVE-023	Unrestricted-State-Variable-Writing	Language	High
TVE-024	Arbitrary-Jump-Function-Type-Variable	Language	Medium
TVE-025	State-Variable-Access-Permissions-Default	Language	Info
TVE-026	Variable-Classification-And-Sorting	Business	High
TVE-027	Dangerous-State-Variable-Shadowing	Language	High
TVE-028	Modifier-To-Modify-State-Variables	Language	High
TVE-029	There-Are-External-Calls-In-The-Modifier	Language	High
TVE-030	Incorrect-Modifier	Language	Low
TVE-031	Multiple-Constructor-Schemes	Language	High
TVE-032	Reused-Base-Constructors	Language	Medium
TVE-033	Void-Constructor	Language	Low
TVE-034	Incorrect-Constructor-Name	Language	Low
TVE-035	Suicidal	Language	High
TVE-036	Fallback-And-Receive()	Language	High
TVE-037	Function-Initializing-State	Language	Info

TVE-038	Unimplemented-Functions	Language	Info
TVE-039	Public-Function-That-Could-Be-Declared-External	Business	Info
TVE-040	Function-Default-Permissions	Language	Info
TVE-041	Unprotected-Withdraw-Function	Language	High
TVE-042	Unchecked-Send	Language	Medium
TVE-043	Unchecked-Transfer	Language	High
TVE-044	Missing-Events-Access-Control	Language	Low
TVE-045	Missing-Events-Arithmetic	Language	Low
TVE-046	Unindexed-Erc20-Event-Parameters	Language	Info
TVE-047	Incorrect-Erc20-Interface	Business	Medium
TVE-048	Incorrect-Erc721-Interface	Business	Medium
TVE-049	Erc20-Approve()-Race-Condition	Language	High
TVE-050	Costly-Operations-Inside-A-Loop	Language	Info
TVE-051	Calls-Inside-A-Loop	Language	Low
TVE-052	Unchecked-Low-Level-Calls	Language	Medium
TVE-053	Low-Level-Calls	Language	Info
TVE-054	Controlled-Delegatecall	EVM	High
TVE-055	Message-Call-With-Hard-Coded-Gas-Number	EVM	Low
TVE-056	Public-Mappings-With-Nested-Variables	Language	High
TVE-057	Deletion-On-Mapping-Containing-A-Structure	Language	Medium

TVE-058	Functions-That-Send-Ether-To-Arbitrary-Destinations	Language	High
TVE-059	Missing-Zero-Address-Validation	Language	Low
TVE-060	Critical-Address-Change	Language	Info
TVE-061	Signature-Replay	EVM	High
TVE-062	Lack-Of-Protection-From-Signature-Replay-Attacks	EVM	High
TVE-063	Redundant-Statements	Language	Info
TVE-064	Unreached-Code	Language	Info
TVE-065	Code-That-Does-Not-Achieve-The-Desired-Effect	Language	Low
TVE-066	Weak-Prng	Blockchain	High
TVE-067	Block-Timestamp	Blockchain	Low
TVE-068	Block-Values-As-Time-Proxies	Blockchain	High
TVE-069	Missing-Inheritance	Language	Info
TVE-070	Incorrect-Order-Of-Inheritance	Language	Low
TVE-071	Whether-To-Inherit	Business	Low
TVE-072	Boolean-Equality	Language	Info
TVE-073	Misuse-Of-A-Boolean-Constant	Language	Medium
TVE-074	Tautology-Or-Contradiction	Language	Medium
TVE-075	Dangerous-Strict-Equalities	Language	Medium

TVE-076	Dangerous-Unary-Expressions	Language	Low
TVE-077	Assert-State-Change	Language	Info
TVE-078	Dangerous-Usage-Of-Tx.Origin	Language	Medium
TVE-079	Unexpected-Ether-And-This.Balance	Language	Medium
TVE-080	Integer-Overflow	Language	High
TVE-081	Divide-Before-Multiply	Language	Medium
TVE-082	Too-Many-Digits	Language	Info
TVE-083	Dirty-High-Order-Bits	Language	Low
TVE-084	Modifying-Storage-Array-By-Value	Language	High
TVE-085	Array-Length-Assignment	Language	High
TVE-086	Incorrect-Shift-In-Assembly	Language	High
TVE-087	Name-Reused	Language	High
TVE-088	Right-To-Left-Override-Character	Language	High
TVE-089	Unprotected-Upgradeable-Contract	Language	High
TVE-090	Contracts-That-Lock-Ether	Business	Medium
TVE-091	Unused-Return	Business	Medium
TVE-092	Assembly-Usage	Language	Info
TVE-093	Deprecated-Standards	Language	Info
TVE-094	Conformance-To-Solidity-Naming-Conventions	Language	Info
TVE-095	Hash-Collision-With-Multiple-Variable-Length-Parameters	EVM	High

TVE-096	Lack-Of-Proper-Signature-Verification	EVM	High
TVE-097	Insufficient-Gas	EVM	Low
TVE-098	Private-On-Chain-Data	Business	Low
TVE-099	Condition-Violation	Language	Low
TVE-100	Write-After-Write	Language	Medium
TVE-101	Incorrect-Access-Control	Language	High
TVE-102	Transaction-Order-Dependence	Blockchain	High
TVE-103	Contract-Check	Language	Low
TVE-104	Deprecated-Keywords	Language	news
TVE-105	Unprotected-Initializer-In-Agent-Based-Upgradable-Contract	Business	High
TVE-106	Initialize-The-State-Variables-In-The-Agent-Based-Upgradable-Contract	Business	High
TVE-107	Import-Agent-Based-Upgradable-Contracts	Business	High
TVE-108	Avoid-Using-Selfdestruct-Or-Delegatecall-In-Proxy-Based-Upgradable-Contracts	Business	High
TVE-109	State-Variables-In-Agent-Based-Upgradable-Contracts	Business	High
TVE-110	Function-Id-Collision-Between-Agents/Contracts-In-Agent-Based-Upgradable-Contracts	Business	High
TVE-111	Functions-Shadowing	Business	High

TVE-112	The-Initialization-Function-Is-Called-Multiple-Times	Business	High
TVE-113	The-Initialization-Of-The-Proxy-Contract-Is-Not-Called	Business	Low
TVE-114	Combine-Business-Checks-That-Must-Be-Initialized-During-Deployment	Business	Low
TVE-115	Combined-Business-Inspection-Must-Be-Initialized	Business	Low
TVE-116	Check-Whether-The-Initialization-Function-Confirms-To-Openzeppelin	Business	Low
TVE-117	Variables-That-Should-Not-Be-Constant	Language	Low
TVE-118	Initialize-Functions-Are-Not-Called	Language	Low
TVE-119	Initializer()-Is-Not-Called	Language	Low
TVE-120	Incorrect-Variables-With-The-V2	Language	Low
TVE-121	Incorrect-Variables-With-The-Proxy	Language	Low
TVE-122	State-Variable-Initialized	Language	Low
TVE-123	Variables-That-Should-Be-Constant	Language	Low
TVE-124	Extra-Variables-In-The-Proxy	Language	Low
TVE-125	Missing-Variables	Language	Low
TVE-126	Extra-Variables-In-The-V2	Language	Low
TVE-127	Initializable-Is-Not-Inherited	Language	Low

TVE-128	Initializable-Is-Missing	Language	Low
TVE-129	Initialize-Function	Language	Low
TVE-130	Initializer()-Is-Missing	Language	High
TVE-131	Abiencoderv2-Array	Language	High
TVE-132	Storage-Type-Signed-Integer-Array-Error	Language	High
TVE-133	Enumeration-Conversion	Language	Medium
TVE-134	Constant-Function-Using-Assembly-Code	Language	Medium
TVE-135	Constant-Function-To-Modify-State-Variables	Language	Medium
TVE-136	Uninitialized-Function-Pointer-In-The-Construct or	Language	Low
TVE-137	Pre-Declared-Usage-Of-Local-Variables	Language	Low
TVE-138	Implicit-Constructor-Callvalue-Check	Language	Medium
TVE-139	Incorrect-Event-Signature-In-Library	EVM	Low
TVE-140	Call-An-Uninitialized-Function-Pointer-In-The-C onstructor	Language	Low
TVE-141	Dynamic-Constructor-Parameters-Are-Abienco derv2	Language	Low
TVE-142	Storage-Array-Of-Multi-Slot-Elements-With-Abi encoderv2	Language	Low
TVE-143	Use-Abiencoderv2-To-Read-The-Calldata-Struct ure-Containing-Static-Size-And-Dynamic-Encod	Language	Low

	ing-Members		
TVE-144	Package-Storage-Using-Abiencoderv2	Language	Low
TVE-145	Incorrect-Loading-Using-Yul-Optimizer-And-Abiencoderv2	Language	Low
TVE-146	Use-Abiencoderv2-To-Dynamically-Encode-Base-Type-Array-Slices	Language	Low
TVE-147	When-Using-Abiencoderv2-The-Formatting-Process-Lacks-Escaping	Language	Low
TVE-148	Double-Shift-Overflow	Language	High
TVE-149	Incorrect-Byte-Instruction-Optimization	Language	Low
TVE-150	Use-Yul-Optimizer-To-Remove-Necessary-Assignments	Language	Low
TVE-151	Private-Method-Is-Overridden	Language	Low
TVE-152	Multi-Stack-Slot-Component-Of-Tuple-Assignment	Language	Low
TVE-153	Dynamic-Array-Cleanup	Language	Low
TVE-154	Empty-Byte-Array-Copy	Language	Low
TVE-155	Memory-Array-Creation-Overflow	Language	Low
TVE-156	Calldata-Using-For	Language	Low
TVE-157	Free-Function-Redefinition	Language	Low
TVE-158	Token-Standard	Business	Low

TVE-159	Asset-Lock	Business	High
TVE-160	Address-Check	Business	High
TVE-161	Community-Governance	Business	High
TVE-162	Flash-Loan	Business	High
TVE-163	Price-Prediction-Machine	Business	High
TVE-164	Minting-And-Burning	Business	High
TVE-165	Exchange-Business	Business	High
TVE-166	Liquidity-Mining	Business	High
TVE-167	Lending-Business	Business	High
TVE-168	Aggregate-Revenue	Business	High
TVE-169	Single-Currency-Pledge	Business	High
TVE-170	Referral-Reward	Business	High
TVE-171	Cross-Platform-Trading	Business	High
TVE-172	Standard-Library-Functions	Business	High

Appendix B: Vulnerability Severity Classification

The nature of vulnerabilities is unintentional and unexpected security flaws or risks, which can be divided into four threat levels: High, Medium, Low and Info. The classification is mainly based on the impact, likelihood of utilization and other factors.

The impact is defined mainly according to the three dimensions of confidentiality, integrity and availability;

The likelihood of utilization is defined mainly according to three dimensions: attack vector, attack complexity and authentication.

Impact \ Likelihood	critical	high	medium	low
low	■ High	■ High	■ Medium	■ Low
medium	■ High	■ Medium	■ Low	■ Low
high	■ Medium	■ Low	■ Low	■ Info
Ex-high	■ Low	■ Low	■ Info	■ Info

Disclaimer

SharkTeam has tried the best to ensure the accuracy and reliability of the content when writing this report, but SharkTeam will not be responsible for the loss and damage caused by the omission, inaccuracy or error in this report. The safety audit analysis and other contents of this report are based on the materials provided by the project team. This audit only focuses on the audit items provided in this report, and other unknown security vulnerabilities are not within the scope of this audit. SharkTeam cannot determine the security status of facts that appear or exist after the report. SharkTeam is not responsible for the background or other circumstances of the project.

The content, services and any resources involved in this report cannot be used as the basis for any form of investment, taxation, law, and supervision, and there is no relevant responsibility.

About SharkTeam

SharkTeam's vision is to secure the Web3 world. The team consists of experienced security professionals and senior researchers from all over the world, proficient in the underlying theory of blockchain and smart contracts, providing services including smart contract auditing, on-chain analysis, emergency response, etc. We have established long-term partnerships with key players in various areas of the blockchain ecosystem, such as Polkadot, Moonbeam, polygon, OKC, Huobi Global, imToken, ChainIDE, etc.

We implement almost 200 auditing contents that cover four aspects: high-level language layer, virtual machine layer, blockchain layer, and business logic layer, ensuring that smart contracts are completely guaranteed and Safe.



SharkTeam

In Math, We Trust!



<https://sharkteam.org>



<https://twitter.com/sharkteamorg>



<https://discord.com/invite/jGH9xXCjDZ>



<https://t.me/sharkteamorg>