smart state

Web3 security easier than ever

XANA

XANA Jetton

Smart contract audit report

October 03, 2024

Table of contents

Table of contents 2

Summary 3

Disclaimer 3

Methodology 4

Vulnerabilities found by type 5

XANA-minter.fc contract methods analysis 6

Verification checksums 8

Project evaluation 9

Contact information 10

Summary

XANA is a leading AI-driven Metaverse and SocialFi platform, ranking as Japan's number one in the Web3 Metaverse space. XANA ecosystem integrates the power of AI, Metaverse, and Web3 technologies. XANA enables users to craft their own AI-powered avatars, immerse themselves in a dynamic universe with a user-generated content, and participate in an array of engaging Web3.0 games.

This audit encompasses the examination of smart contracts of the XANA token on the TON protocol.

Disclaimer

This is a final public security audit report version that might not include vulnerabilities that might have been found and addressed during the audit process.
An audit does not provide any warranties regarding the code security. We presume that a single audit cannot be considered totally sufficient and always recommend several independent audits and a public bug bounty program to ensure code security.
Please do not consider this report as investment and / or financial advice of any kind.

Methodology

- Manual code analysis
- Best code practices
- FA2 compliance (if applicable)
- Logical bugs & code logic issues
- Error handling issues
- Cryptographic errors
- Protocol and header parsing errors
- Private data leaks
- Unchecked call return method
- Code with no effects
- Unused vars
- Use of deprecated functions
- Authorization issues
- Reentrancy
- Arithmetic overflows / underflows
- Hidden malicious code
- External contract referencing
- Short address/parameter attack

- Uninitialized storage pointers
- Floating points and precision
- Message rebounce
- The order of data import
- Consider the case where a message fails
- Cost refund
- Cell data and storage params
- Security of concurrent message calls and locks
- Access control is enforced properly
- Asynchronous messages do not create race condition
- Address formats handled correctly
- Gas accounting is correct
- Bounced messages are handled correctly
- The funds are reserved correctly
- Function specifiers are correct
- Logic is implemented properly

Vulnerabilities found by type

INFO	0
LOW	0
MEDIUM	0
HIGH	0
CRITICAL	0
Total	0

XANA-minter.fc contract methods analysis:

`(int, slice, cell, cell) load_data()`
Vulnerabilities not detected
Math issues not detected

`() save_data(int total_supply, slice admin_address, cell content, cell jetton_wallet_code)`
Vulnerabilities not detected
Math issues not detected

`() mint_tokens(slice to_address, cell jetton_wallet_code, int amount, cell master_msg)`
Vulnerabilities not detected
Math issues not detected

`() recv_internal(int msg_value, cell in_msg_full, slice in_msg_body)`
Vulnerabilities not detected
Math issues not detected

`(int, int, slice, cell, cell) get_jetton_data()`
Vulnerabilities not detected
Math issues not detected

XANA-minter.fc contract methods analysis:

slice get_wallet_address(slice owner_address)
Vulnerabilities not detected
Math issues not detected

Verification checksums

Contract	Bytecode hash(SHA-256)
XANA-minter.fc	b93612f1f3e3d321e9695f103b1e697e97f5ba5e2da9e519987904fd667fe7ef

Token on block explorer
https://tonviewer.com/EQBQFE7NKgvZAAbPZvCrUD0YDr5vr32OdVA97_VjtRhDViz0

Project evaluation

■■■■■■■■■■

10/10

Get in touch 👋

@smartstatetech

@smartstate

@SmartStateAudit

@smartstatetech

@smartstate.tech

View this report on Smartstate.tech

info@smartstate.tech

smartstate.tech