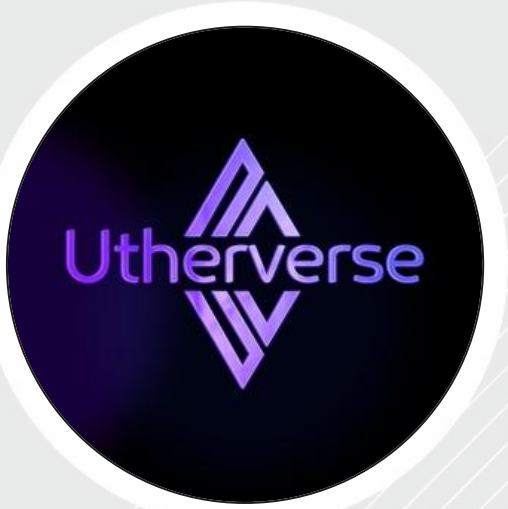
SPYWOLF

Security Audit Report



Audit prepared for

Utherverse: Stake Program

Completed on

October 22, 2024

OVERVIEW

This goal of this report is to review the main aspects of the project to help investors make an informative decision during their research process.

You will find a a summarized review of the following key points:

- ✔ Program’s source code
- ✔ Team transparency and goals
- ✔ Website's age, code, security and UX
- ✔ Whitepaper and roadmap
- ✔ Social media & online presence

The results of this audit are purely based on the team’s evaluation and does not guarantee nor reflect the projects outcome and goal

- SPYWOLF Team -

TABLE OF CONTENTS

UTHERVERSE

PROJECT DESCRIPTION

Utherverse is not just another player in the metaverse space – we are the pioneers, with over 15 years of experience in building successful virtual economies and communities. Our track record speaks for itself, but our vision for the future is what truly sets us apart. By harnessing the immense potential of web3, blockchain, and AI, we are creating a metaverse that is unmatched in its immersion, adaptability, and profitability.

Release Date: Launches in October, 2024

Category: Staking

PROGRAM INFO

Token Name
Stake Program

Symbol

Contract Address
4JDLHn7pSyjbnzcMXLg5NdLRpNLiVNU8B2Z6CgzVAwSz

Network
Solana

Language
Rust

Deployment Date
Sep 17, 2024

Contract Type
Staking

Total Supply
10,000,000,000

Status
devnet

TAXES

Buy Tax
0

Sell Tax
0

*This type of program does not have taxes

Our Program Review Process

The contract review process pays special attention to the following:

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

Blockchain security tools used:

- Solana Program Library (SPL)
- Manual Auditing / Sec3 / Neodyme
- Rust Compiler
- Anchor Framework

VULNERABILITY ANALYSIS

ERRORS FOUND

Initialize: Initializes the staking pool with parameters.

☐ Low Risk

Lack of Input Validation: The function does not validate the input parameters (lock_time, apy, apy_denominator, roi_type). For example, if apy_denominator is zero, it could lead to a division by zero error in later calculations.

Potential Misconfiguration: If the wrong admin address is set, it could lead to unauthorized access.

```rust
pub fn initialize(
    ctx: Context<Initialize>,
    lock_time: u64,
    apy: u64,
    apy_denomiator: u64,
    roi_type: u64,
) -> Result<()> {
    let pool_info = &mut ctx.accounts.pool_info;

    pool_info.admin = ctx.accounts.admin.key();
    pool_info.token_vault = ctx.accounts.token_vault_account.key();
    pool_info.lock_time = lock_time;
    pool_info.apy = apy;
    pool_info.apy_denominator = apy_denomiator;
    pool_info.roi_type = roi_type; // 0-> Daily, 1-> Weekly, 2-> Monthly
    pool_info.token = ctx.accounts.mint.key();

    Ok(())
}
```

- Recommendation:
  - Add security validation to the input parameters

VULNERABILITY ANALYSIS

ERRORS FOUND

stake: Allows users to stake tokens

☐ Low Risk

Reentrancy Attack: The function transfers tokens before updating the state variables. An attacker could exploit this by calling stake recursively before the state is updated.

Insufficient Checks on stake_counter: The function does not ensure that the stake_couter is unique or valid, which could lead to issues with tracking stakes.

```
stake_info.stake_seed = stake_counter;
```

- Recommendation:
  - Add checks to stake_counter if it's unique

VULNERABILITY ANALYSIS

ERRORS FOUND

destake: Allows users to unstake their tokens

☐ Low Risk

Reward Calculation Logic: The reward calculation loop could potentially lead to excessive gas consumption if not bounded properly.

- Recommendation:
  - Lock Period Enforcement: Ensure that the lock period is strictly enforced and consider edge cases where the clock might be manipulated.
  - State Resetting: Reset all relevant state variables after unstaking to prevent stale data from being used in future transactions.
  - Reward Calculation Logic: Ensure that reward calculations are bounded to avoid excessive gas consumption and potential infinite loops.

VULNERABILITY ANALYSIS

ERRORS FOUND

calculate_rewards: Calculates rewards based on staked amounts

☐ Low Risk

Division by Zero Risk: Similar to initialize, if pool_info.apy_denominator is zero, this will cause a panic at runtime

```
let reward_rate = stake_info.staked_amount * pool_info.apy
    / pool_info.apy_denominator
    / constants::SLOTS_PER_YEAR;
```

- Recommendation:
  - Input Validation: Validate that the stake amount is positive before performing calculations.
  - Handle Division by Zero: Ensure that `apy_denominator` is not zero before performing division.

VULNERABILITY ANALYSIS

ERRORS FOUND

claim_rewards: Allows users to claim their rewards

Low Risk

Timing Issues: If multiple claims are made simultaneously, it could lead to inconsistent states due to race conditions.

Unclaimed Rewards Reset Logic: Resetting unclaimed rewards without proper checks can lead to loss of rewards if the logic fails.

- Recommendation:
  - Timing Checks: Ensure that sufficient time has passed before allowing claims and validate that the user has not already claimed rewards.
  - Reentrancy Protection: Similar to other functions, ensure state changes are made before external calls.

VULNERABILITY ANALYSIS

ERRORS FOUND

restake_rewards: Allows users to restake their rewards

☐ Low Risk

Similar Timing and State Issues as in claim_rewards: The same vulnerabilities regarding race conditions and state management apply here.

- Recommendation:
  - Similar Checks as `claim_rewards`: Implement similar checks for timing, reentrancy, and state validation as done in the `claim_rewards` function.

VULNERABILITY ANALYSIS

ERRORS FOUND

update_pool_info: updates pool parameters by an admin

Passed

VULNERABILITY ANALYSIS

ERRORS FOUND

admin_withdraw: Allows admin to withdraw funds from treasury

Passed

SPYWOLF

CRYPTO SECURITY

ABOUT US

We are a growing crypto security agency offering audits, KYCs and consulting services for some of the top names in the crypto industry.

- **OVER 700 SUCCESSFUL CLIENTS**
- **MORE THAN 1000 SCAMS EXPOSED**
- **MILLIONS SAVED IN POTENTIAL FRAUD**
- **PARTNERSHIPS WITH TOP LAUNCHPADS, INFLUENCERS AND CRYPTO PROJECTS**
- **CONSTANTLY BUILDING TOOLS TO HELP INVESTORS DO BETTER RESEARCH**

Audits | KYCs | dApps
Contract Development

To hire us, reach out to
contact@spywolf.co or
t.me/joe_SpyWolf

FIND US ONLINE

SPYWOLF.CO

@SPYWOLFNETWORK

@SPYWOLFNETWORK

Disclaimer

This report shows findings based on our limited project analysis, following good industry practice from the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, overall social media and website presence and team transparency details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report.
While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the disclaimer below – please make sure to read it in full.

DISCLAIMER:
By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice.
No one shall have any right to rely on the report or its contents, and SpyWolf and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (SpyWolf) owe no duty of care towards you or any other person, nor does SpyWolf make any warranty or representation to any person on the accuracy or completeness of the report.
The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and SpyWolf hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, SpyWolf hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against SpyWolf, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report. The analysis of the security is purely based on the smart contracts, website, social media and team.
No applications were reviewed for security. No product code has been reviewed.