SOLIDIFIED

Audit Report for Paraswap - September 01, 2021

Summary

Audit Report prepared by Solidified covering the Paraswap PSP staking smart contract.

Process and Delivery

Threw (3) independent Solidified experts performed an unbiased and isolated audit of the code. The debrief was held on 1 September 2021.

Audited Files

The source code has been supplied in the form of a GitHub repository:

https://github.com/BlockzeroLabs/vortex-contracts

Commit number: `82c7cc84df342948532a4af8009dec1dd5e10b13`

The scope of the audit was limited to the following files:

```
contracts/
├── staking
│   └── SPSP.sol
└── test
    └── TestToken.sol
```

Intended Behavior

The smart contracts implement a staking solution that rewards stakers with PSP tokens that are added to the contract by Paraswap. Stakers are issued with sPSP tokens and these can be exchanged on unstaking for the proportional share of PSP tokens in the contract. The staking rewards, thus depending on Paraswap adding PSP to the pool.

Code Complexity and Test Coverage

Smart contract audits are an important step to improve the security of smart contracts and can find many issues. However, auditing complex codebases has its limits and a remaining risk is present (see disclaimer).

Users of a smart contract system should exercise caution. In order to help with the evaluation of the remaining risk, we provide a measure of the following key indicators: **code complexity**, **code readability**, **level of documentation**, and **test coverage**.

Note, that high complexity or lower test coverage does equate to a higher risk. Certain bugs are more easily detected in unit testing than a security audit and vice versa. It is, therefore, more likely that undetected issues remain if the test coverage is low or non-existent.

Criteria	Status	Comment
Code complexity	Low	-
Code readability and clarity	High	-
Level of Documentation	High	-
Test Coverage	High	-

Issues Found

Solidified found that the Paraswap contracts contain 1 warning, no critical issues, 2 major issues, 1 minor issue in addition to 3 informational notes.

We recommend all issues are amended, while the notes are up to the team's discretion, as they refer to best practices.

Issue #	Description	Severity	Status
1	Staking rewards are entirely dependent on funds being transferred to the contract regularly	Warning	
2	Initial rewards are assigned entirely to first staker	Major	Pending
3	Contract rewards are susceptible to front running and/or MEV (Miner Extractable Value)	Major	Pending
4	ERC-20 return values ignored	Minor	Pending
5	User indexes and withdrawal index can be unsigned integers	Note	-
6	Consider providing a function for retrieving unlocked IDs	Note	-
7	Miscellaneous notes	Note	-

Warnings

1. Staking rewards are entirely dependent on funds being transferred to the contract regularly

The contract depends on external sources for the rewards. Should Paraswap fail to feed additional PSP tokens into the contract, no rewards will be available on unstaking.

Recommendation
Document this constraint to users so that they can pre-calculate worst-case rewards and make the refunding policy clear.

Critical Issues

No critical issues were found.

Major Issues

2. Initial rewards are assigned entirely to first staker

The way the PSPs-PSP ratio is calculated when staking means that any PSP rewards seeded in the contract before staking commences are automatically assigned to the first staker. This means that the contract cannot be used for distributing a pre-assigned amount of PSP. Any further rewards added are subject to Paraswap transferring further funds (see warning above).

Recommendation
Consider changing the reward calculation or avoid pre-seeding the reward distribution (in which case the above warning applies).

3. Contract rewards are susceptible to front running and/or MEV (Miner Extractable Value)

Since the rewards are distributed as `PSP` tokens that are directly sent to the contract, anyone can leverage the MEV or front running to mint new `sPSP` tokens right before the rewards are distributed and claim the reward which was meant for the original staker.

Recommendation
Consider changing the way rewards are distributed to a more standard time-based reward system.

Minor Issues

4. ERC-20 return values ignored

The contract ignores the return values of ERC-20 calls. Whilst this is fine for most tokens, including, most likely PSP, some tokens do not revert on error and return false instead. It is generally considered best practice to include checks for this in case the code is reused with incompatible tokens.

Recommendation
Check return values of ERC-20 calls.

Informational Notes

5. User indexes and withdrawal index can be unsigned integers

The indexes used for keeping track of staking and withdrawal indexes per user are of type `int256`. Whilst this is fine, it essentially halves the number of available indexes (which is still a very large number) and is less intuitive.

Recommendation

Consider using `uint256`.

6. Consider providing a function for retrieving unlocked IDs

After calling `leave()`, the only way for users to know their current pending unlocked withdrawal IDs is via checking all the `Unstaked()` logs that were emitted.

Recommendation

Consider both implementing a `findUnlockedIDs()` function and returning the respective withdrawal ID in function `leave()`.

7. Miscellaneous notes

The following are some misc notes that can help improve the code quality and readability.

- The validation `require(request.status == WITHDRAW_STATUS.UNUSED, "Invalid id")` will always return true and can be removed.

Disclaimer

Solidified audit is not a security warranty, investment advice, or an endorsement of Paraswap or its products. This audit does not provide a security or correctness guarantee of the audited smart contract. Securing smart contracts is a multistep process, therefore running a bug bounty program as a complement to this audit is strongly recommended.

The individual audit reports are anonymized and combined during a debrief process, in order to provide an unbiased delivery and protect the auditors of Solidified platform from legal and financial liability.

Solidified Technologies Inc.