SOLIDIFIED

Audit Report for Stakefish - September 20, 2021

Summary

Audit Report prepared by Solidified covering the Stakefish smart contracts.

Process and Delivery

Three (3) independent Solidified experts performed an unbiased and isolated audit of the code below. The final debrief took place on September 20, 2021, and the results are presented here.

Audited Files

The source code has been supplied in a public source code repository:
https://github.com/stakefish/eth2-validation-services-contract/commit/31bd3e263b7355583b7c897582758b74cc9ce5c8

Commit number: `31bd3e263b7355583b7c897582758b74cc9ce5c8`

```
├── StakefishERC20Wrapper.sol
├── StakefishERC721Wrapper.sol
├── StakefishServicesContract.sol
├── StakefishServicesContractFactory.sol
├── interfaces
│   ├── IERC165.sol
│   ├── IERC20.sol
│   ├── IERC721.sol
│   ├── IERC721Receiver.sol
│   ├── IStakefishServicesContract.sol
│   ├── IStakefishServicesContractFactory.sol
│   ├── MockDepositContract.sol
│   └── deposit_contract.sol
└── libraries
    ├── Address.sol
    ├── Initializable.sol
    ├── ProxyFactory.sol
    └── ReentrancyGuard.sol
```

Intended Behavior

Stakefish is a blockchain staking service provider.

Findings

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 not necessarily equate to a higher risk, although certain bugs are more easily detected in unit testing than a security audit and vice versa.

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

SOLIDIFIED

Audit Report for Stakefish - September 20, 2021

Issues Found

Solidified found that the Stakefish contracts contain no critical issues, no major issues, 1 minor issue, and 4 informational notes.

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

Issue #	Description	Severity	Status
1	StakefishERC20Wrapper.sol: Token allowance susceptible to front running	Minor	Pending
2	StakefishServicesContract.sol: Function receive() should revert if state is NotInitialized	Note	-
3	Missing validations	Note	-
4	Some revert error messages are longer than 32 characters	Note	-
5	Miscellaneous Notes	Note	-

Critical Issues

No critical issues have been found.

Major Issues

No major issues have been found.

Minor Issues

1. StakefishERC20Wrapper.sol: Token allowance susceptible to front running

Changing the account allowance through the `approve()` method brings the risk that someone may use both the old and the new allowance by unfortunate transaction ordering. A detailed description of this vulnerability can be found here: https://docs.google.com/document/d/1YLPtQxZu1UAvO9cZ1O2RPXBbT0mooh4DYKjA_jp-RLM

Recommendation

Consider mitigating this race condition by implementing `increaseAllowance` and `decreaseAllowance` functions to update the allowance.

Informational Notes

2. StakefishServicesContract.sol: Function receive() should revert if state is NotInitialized

Recommendation

Consider having function `receive()` revert before the contract is initialized.

3. Missing validations

The contracts in several places do not validate the input parameters. The following are a few such places that requires extra validation

1. `StakefishServicesContractFactory.sol`: `Operator` can unintentionally revoke it's access by calling the `changeOperatorAddress` method with a zero address.
2. `StakefishServicesContractFactory.sol` and `StakefishServicesContract.sol`: Ensure the `commisionRate` is less than the `commisionRateScale`.

Recommendation

Consider adding the recommended validations.

4. Some revert error messages are longer than 32 characters

Revert error messages greater than 32 characters can unnecessarily increase the contract size and can use slightly more gas when reverting.

Recommendation

Consider rewriting error messages to reduce their size to 32 characters.

5. Miscellaneous Notes

- StakefishServicesContract.sol: No parameter validation for function `initialize()`.
- StakefishERC20Wrapper.sol: Functions `mintTo()` and `redeemTo()` error messages refer to `ERC20Wrapper` instead of `StakefishERC20Wrapper`.
- Consider using the standard `onlyInitialized` modifier pattern with all `initialize()` functions.
- Consider fixing spelling mistakes in the comments: seperate → separate, tamplete → template, nonexistent → non-existent.

Audit Report for Stakefish - September 20, 2021

Disclaimer

Solidified audit is not a security warranty, investment advice, or an endorsement of Stakefish 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.