CERTIK

Six Network

LP Farm Contracts

Security Assessment

March 29th, 2021

Audited By:
Alex Papageorgiou @ CertiK
alex.papageorgiou@certik.org
Reviewed By:
Camden Smallwood @ CertiK
camden.smallwood@certik.org

Disclaimer

CertiK reports are not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. These reports are not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts CertiK to perform a security review.

CertiK Reports do not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

CertiK Reports should not be used in any way to make decisions around investment or involvement with any particular project. These reports in no way provide investment advice, nor should be leveraged as investment advice of any sort.

CertiK Reports represent an extensive auditing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. CertiK's position is that each company and individual are responsible for their own due diligence and continuous security. CertiK's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

What is a CertiK report?

- A document describing in detail an in depth analysis of a particular piece(s) of source code provided to CertiK by a Client.
- An organized collection of testing results, analysis and inferences made about the structure, implementation and overall best practices of a particular piece of source code.
- Representation that a Client of CertiK has completed a round of auditing with the intention to increase the quality of the company/product's IT infrastructure and or source code.

Overview

Project Summary

Project Name	Six Network - LP Farm Contracts
Description	An LP Farm implementation based on Pancake's farms.
Platform	Ethereum; Solidity, Yul
Codebase	GitHub Repository
Commits	1. a7403569afe57cddf5b969ccd3287fff3604f5c0 2. 1f4e291d04d102508aa84f94a703082df7b0db78 3. a7403569afe57cddf5b969ccd3287fff3604f5c0

Audit Summary

Delivery Date	March 29th, 2021
Method of Audit	Static Analysis, Manual Review
Consultants Engaged	1
Timeline	March 26th, 2021 - March 29th, 2021

Vulnerability Summary

Total Issues	2
🔴 **Total Critical**	0
🟠 **Total Major**	1
🟡 **Total Medium**	0
🔵 **Total Minor**	1
🟢 **Total Informational**	0

Executive Summary

We were tasked with auditing the codebase of the Definix LP farm implementations that are based on PancakeSwap's farms in turn based on SushiSwap's MasterChef farm.

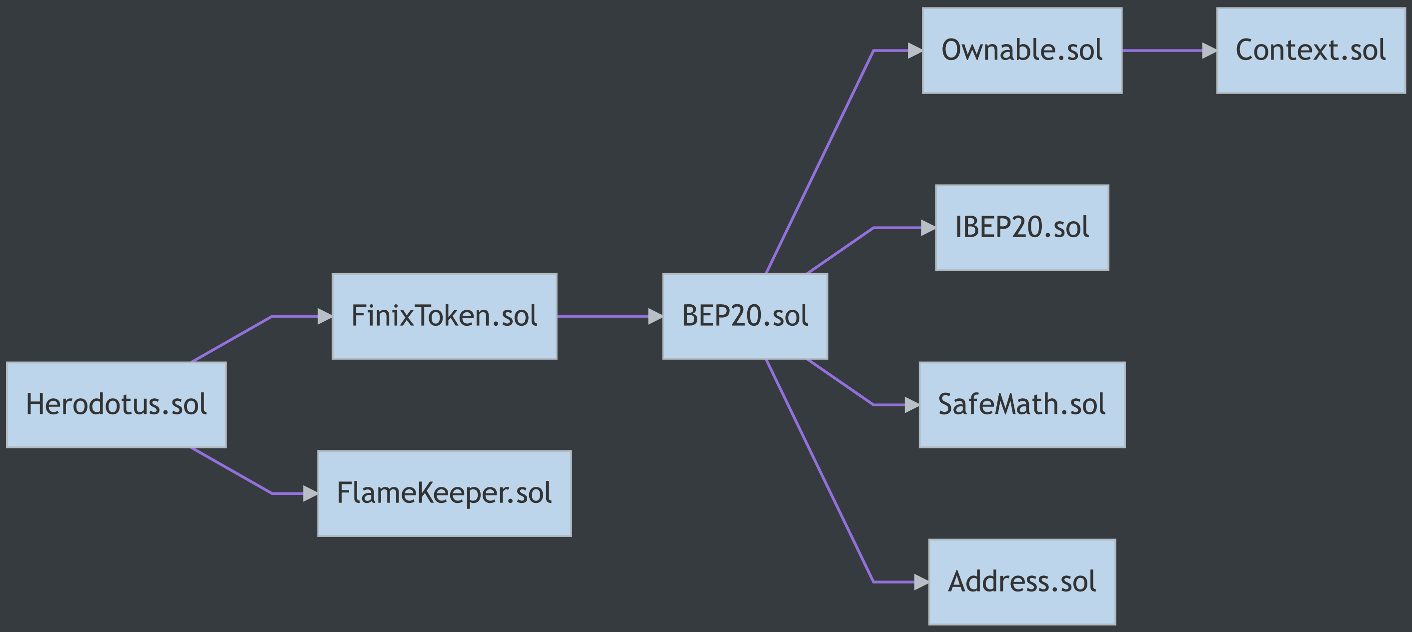
The Definix team removed the incentivization of the first pool of the farm that was meant to incentivize users to deposit on the `SYRUP` token of the PancakeSwap implementation and instead opted to treat it as a normal pool along with the others.

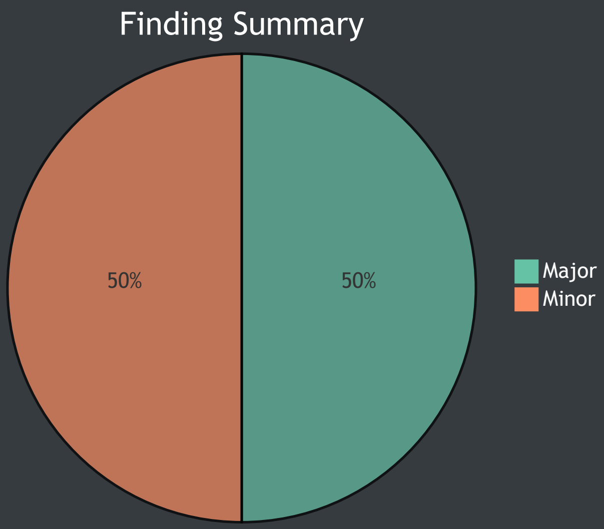
A vulnerability that was identified in the PancakeSwap project remains within the Definix implementation whereby `FLAME` tokens can be infinitely minted. We advise this exploit to be fixed to ensure that the `FLAME` token accrues value and does not become worthless.

Files In Scope

ID	Contract	Location
BSG	BnbStaking.sol	contracts/BnbStaking.sol
FTN	FinixToken.sol	contracts/FinixToken.sol
FKR	FlameKeeper.sol	contracts/FlameKeeper.sol
HER	Herodotus.sol	contracts/Herodotus.sol
SCF	SousChef.sol	contracts/SousChef.sol
TIM	Timelock.sol	contracts/Timelock.sol
VFT	VerifyFinixToken.sol	contracts/VerifyFinixToken.sol
MIG	Migrations.sol	contracts/libs/Migrations.sol
MUL	Multicall.sol	contracts/libs/Multicall.sol
PVP	PancakeVoteProxy.sol	contracts/libs/PancakeVoteProxy.sol
WBN	WBNB.sol	contracts/libs/WBNB.sol
CON	Context.sol	contracts/pancake-swap-lib/contracts/GSN/Context.sol
OWN	Ownable.sol	contracts/pancake-swap-lib/contracts/access/Ownable.sol
SMH	SafeMath.sol	contracts/pancake-swap-lib/contracts/math/SafeMath.sol
ADD	Address.sol	contracts/pancake-swap-lib/contracts/utils/Address.sol
BEP	BEP20.sol	contracts/pancake-swap-lib/contracts/token/BEP20/BEP20.sol
IBE	IBEP20.sol	contracts/pancake-swap-lib/contracts/token/BEP20/IBEP20.sol

File Dependency Graph





Finding Summary

Manual Review Findings

ID	Title	Type	Severity	Resolved
HER-01	Potential for Unlimited Minting	Logical Issue	● Major	✓
HER-02	Checks-Effects-Interactions Pattern	Logical Issue	● Minor	ⓘ

HER-01: Potential for Unlimited Minting

Type	Severity	Location
Logical Issue	● Major	Herodotus.sol L263-L280, L283-L290

Description:

The `FLAME` token is minted whenever deposits are made to the first pool defined which is meant to represent the Finix token staking pool where users are able to mint flame tokens and consequently burn them on exit.

Recommendation:

The `emergencyWithdraw` function enables the users to bypass the burning mechanism thus enabling them to `enterStaking` and `emergencyWithdraw` repeatedly, minting unlimited tokens. We advise an additional logic path to be coded in the `emergencyWithdraw` function that burns the corresponding amount of `FLAME` tokens if the `_pid` is equal to `0` to ensure users aren't able to exploit this functionality.

Alleviation:

The amount of `FLAME` tokens are properly burned on an `emergencyWithdraw` thus alleviating this exhibit.

HER-02: Checks-Effects-Interactions Pattern

Type	Severity	Location
Logical Issue	● Minor	Herodotus.sol L283-L290

Description:

The `emergencyWithdraw` function performs a `safeTransfer` invocation with an input variable that is zeroed out from `storage` after the external call has concluded.

Recommendation:

We advise the zeroing out of variables to be re-ordered before the external call ensuring that no type of re-entrancy attack can occur even in the case of an EIP-777 token or generally a token that informs the recipient of a transfer.

Alleviation:

The Checks-Effects-Interactions pattern has not been applied in the latest version of the codebase.

Appendix

Finding Categories

Logical Issue

Logical Issue findings are exhibits that detail a fault in the logic of the linked code, such as an incorrect notion on how `block.timestamp` works.