GUARDIAN

Baseline

BToken

Security Assessment

February 25th, 2025

Summary

Audit Firm Guardian

Prepared By Daniel Gelfand, Osman Ozdemir, Nicholas Chew

Client Firm Baseline

Final Report Date February 25, 2025

Audit Summary

Baseline engaged Guardian to review the security of their BToken Updates. From the 24th Of October to the 27th of October, a team of 3 auditors reviewed the source code in scope. All findings have been recorded in the following report.

For a detailed understanding of risk severity, source code vulnerability, and potential attack vectors, refer to the complete audit report below.

🔗 Blockchain network: **Blast**

✅ Verify the authenticity of this report on Guardian's GitHub: https://github.com/guardianaudits

📊 Code coverage & PoC test suite: https://github.com/GuardianAudits/Baseline-Perps

Table of Contents

Project Information

Smart Contract Risk Assessment

Addendum

Project Overview

Project Summary

Project Name	Baseline
Language	Solidity
Codebase	https://github.com/0xBaseline/baseline-v2
Commit(s)	Initial commit: a20a6625f58e1e54f06ca92d2a4cd5f4d6c40c6 Final commit: 4e6670a40af2c48d40869df84722a566d4a949c8

Audit Summary

Delivery Date	February 25, 2025
Audit Methodology	Static Analysis, Manual Review, Test Suite, Contract Fuzzing

Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Partially Resolved	Resolved
● Critical	0	0	0	0	0	0
● High	0	0	0	0	0	0
● Medium	4	0	0	2	0	2
● Low	4	0	0	2	0	2

Audit Scope & Methodology

Vulnerability Classifications

Severity	Impact: *High*	Impact: *Medium*	Impact: *Low*
Likelihood: *High*	🔴 Critical	🟠 High	🟡 Medium
Likelihood: *Medium*	🟠 High	🟡 Medium	🟢 Low
Likelihood: *Low*	🟡 Medium	🟢 Low	🟢 Low

Impact

High Significant loss of assets in the protocol, significant harm to a group of users, or a core functionality of the protocol is disrupted.

Medium A small amount of funds can be lost or ancillary functionality of the protocol is affected. The user or protocol may experience reduced or delayed receipt of intended funds.

Low Can lead to any unexpected behavior with some of the protocol's functionalities that is notable but does not meet the criteria for a higher severity.

Likelihood

High The attack is possible with reasonable assumptions that mimic on-chain conditions, and the cost of the attack is relatively low compared to the amount gained or the disruption to the protocol.

Medium An attack vector that is only possible in uncommon cases or requires a large amount of capital to exercise relative to the amount gained or the disruption to the protocol.

Low Unlikely to ever occur in production.

Audit Scope & Methodology

Methodology

Guardian is the ultimate standard for Smart Contract security. An engagement with Guardian entails the following:

- Two competing teams of Guardian security researchers performing an independent review.
- A dedicated fuzzing engineer to construct a comprehensive stateful fuzzing suite for the project.
- An engagement lead security researcher coordinating the 2 teams, performing their own analysis, relaying findings to the client, and orchestrating the testing/verification efforts.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross-referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts. Comprehensive written tests as a part of a code coverage testing suite.
- Contract fuzzing for increased attack resilience.

Findings & Resolutions

ID	Title	Category	Severity	Status
M-01	Lack Of Policy Permissions	Access Control	● Medium	Acknowledged
M-02	Incorrect BPOOL Locked State	Logical Error	● Medium	Resolved
M-03	Exit Loop Before Borrow Operation	Logical Error	● Medium	Resolved
M-04	probabilityDenominator Increased If No Swap	Logical Error	● Medium	Acknowledged
L-01	Lack Of Validation In setController	Warning	● Low	Acknowledged
L-02	Remove Console2 Imports	Informational	● Low	Resolved
L-03	Misleading Developer Comments	Informational	● Low	Resolved
L-04	Unexpected Behavior During High Premiums	Logical Error	● Low	Acknowledged

M-01 | Lack Of Policy Permissions

Category	Severity	Location	Status
Access Control	● Medium	BPOOL.v1.sol: 254	Acknowledged

Description

`migrateBToken` function in the BPOOL module is a `permissioned` function and it should be called via policies. However, none of the policies have permission to call this function.

Recommendation

Consider which policy is expected to call this function and add the function's selector to the `requestPermissions` process.

Resolution

Baseline Team: Acknowledged.

M-02 | Incorrect BPOOL Locked State

Category	Severity	Location	Status
Logical Error	● Medium	BPOOL.v1.sol: 75	Resolved

Description

ERC20 features are moved from BPOOL to separate BToken contract with the current update. However, the locked status still remains in the BPOOL contract, in addition to the BToken contract, creating an asymmetry.

The setTransferLock function correctly updates the locked status of the BToken contract, but there is no mechanism to update the locked status in BPOOL. Since BPOOL.locked is set to true in the constructor, it will always appear locked.

This will lead to discrepancies for integrators who read BPOOL.locked instead of BPOOL.bToken.locked.

Recommendation

Remove the locked from the BPOOL and use it only from the BToken.

Resolution

Baseline Team: The issue was resolved in commit 9e4a37e.

M-03 | Exit Loop Before Borrow Operation

Category	Severity	Location	Status
Logical Error	● Medium	Afterburner.sol: 254	Resolved

Description

In the function `_loop`, an early exit occurs if `minimumCollateral` is hit. This is called after the borrow operation.

However, if `_bAssetsIn` is too small, the borrow operation could revert as no new principal is transferred out. This would result in the entire `reheat` operation reverting.

Recommendation

The early exit for `minimumCollateral` should be done before the borrow operation.

Resolution

Baseline Team: The issue was resolved in commit 4e6670a.

M-04 | probabilityDenominator Increased If No Swap

Category	Severity	Location	Status
Logical Error	● Medium	Afterburner.sol: 218	Acknowledged

Description

In the reheat function, the probability denominator is incremented with each successful hit to make future hits less likely.

However, when reserveSize = 0, no swap occurs despite the successful hit, but the probability denominator is still incremented.

This reduces the likelihood of true hits (where actual swaps occur) since hits that result in no swaps still affect the probability, making legitimate hits less frequent.

Recommendation

Consider incrementing the probability denominator only if a swap occurs.

Resolution

Baseline Team: Acknowledged.

L-01 | Lack Of Validation In setController

Category	Severity	Location	Status
Warning	● Low	BToken.sol	Acknowledged

Description

The setController is a critical function that hands control of mint/burn ability to a new address. Given the importance of the function, consider validating the new controller address to avoid handing control to an unintended address.

Recommendation

Validate that _controller is not a zero address and consider implementing a two-step handover process.

Resolution

Baseline Team: Acknowledged.

L-02 | Remove Console2 Imports

Category	Severity	Location	Status
Informational	● Low	Global	Resolved

Description

console2 imports were found in:
- LOOPS.v1.sol
- BaselineInit.sol
- LoopFacility.sol
- MarketMaking.sol

Recommendation

Remove the import statements.

Resolution

Baseline Team: The issue was resolved in commit 9e4a37e.

L-03 | Misleading Developer Comments

Category	Severity	Location	Status
Informational	● Low	MarketMaking.sol: 281, 513 - LOOPS.v1.sol: 189	Resolved

Description

Comments on the function drop and _decrementSweepTick indicate that tick will be moved exactly one tick spacing lower. However, with the revised _getDecrementedSweepTick, it is possible to move the tick by more than one tick spacing.

Additionally, the “transfer any surplus collateral back to the bAsset contract” comment in the LOOPS contract is incorrect, and it should be “transfer any surplus collateral back to the BPOOL contract”.

Recommendation

Update developer comments.

Resolution

Baseline Team: The issue was resolved in commit 4e6670a.

L-04 | Unexpected Behavior During High Premiums

Category	Severity	Location	Status
Logical Error	● Low	Afterburner.sol	Acknowledged

Description

When the premium between the active price and BLV is too high, no reserves are swapped during reheat due to capital inefficiency.

However, bAssets are still converted to reserves via borrowing from the CreditFacility and defaulting on self, which burns bAsset collateral.

Burning tokens can create upward pressure on the price, further increasing the premium, which is undesirable.

Recommendation

If the premium is too high, consider gracefully exiting without performing any borrowing or swapping actions.

Resolution

Baseline Team: Acknowledged.

Disclaimer

This report is not, nor should be considered, an “endorsement” or “disapproval” of any particular project or team. This report is 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 Guardian to perform a security assessment. This report does 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.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing 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. Guardian’s position is that each company and individual are responsible for their own due diligence and continuous security. Guardian’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.

The assessment services provided by Guardian is subject to dependencies and under continuing development. You agree that your access and/or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon, multiple layers of third-parties.

Notice that smart contracts deployed on the blockchain are not resistant from internal/external exploit. Notice that active smart contract owner privileges constitute an elevated impact to any smart contract’s safety and security. Therefore, Guardian does not guarantee the explicit security of the audited smart contract, regardless of the verdict.

About Guardian Audits

Founded in 2022 by DeFi experts, Guardian Audits is a leading audit firm in the DeFi smart contract space. With every audit report, Guardian Audits upholds best-in-class security while achieving our mission to relentlessly secure DeFi.

To learn more, visit https://guardianaudits.com

To view our audit portfolio, visit https://github.com/guardianaudits

To book an audit, message https://t.me/guardianaudits