GA GUARDIAN

Baseline Markets

YesArena & Afterburner

Security Assessment

June 18th, 2024

Summary

Audit Firm Guardian

Prepared By Owen Thurm, Daniel Gelfand

Client Firm Baseline Markets

Final Report Date June 18th

Audit Summary

Baseline Markets engaged Guardian to review the security of its YesArena game and afterburner updates. From June 12th to June 16th, a team of 2 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

Table of Contents

Project Information

Project Overview 4

Audit Scope & Methodology 5

Smart Contract Risk Assessment

Findings & Resolutions 7

Addendum

Disclaimer 23

About Guardian Audits 24

Project Overview

Project Summary

Project Name	Baseline Markets
Language	Solidity
Codebase	https://github.com/0xBaseline/baseline-v2
Commit(s)	afterburner-updates: adbb0fc7e0494d81c53e8426a02a32cfcc266485 yes-arena: 21eebf3fec8d48cde0b92dcafc9ba33b4687dc5f

Audit Summary

Delivery Date	June 18th, 2024
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	2	1	0	0	0	1
Medium	4	4	0	0	0	0
Low	9	9	0	0	0	0

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
H-01	Invalid Portion Burned	Logical Error	High	Resolved
H-02	YesArena Block Stuffing Attack	Block Stuffing	High	Pending
M-01	claimed Value Not Assigned	Logical Error	Medium	Pending
M-02	Reheat Sniping	Sandwhich Attack	Medium	Pending
M-03	Blast Yields Are Not Configured For YesArena	Configuration	Medium	Pending
M-04	Lacking _buy Slippage Protection	Sandwhich Attack	Medium	Pending
L-01	Lack Of Upgradeability Controls	Suggestion	Low	Pending
L-02	Lacking Rate Validations	Validation	Low	Pending
L-03	YesArena References Old AfterBurner	Configuration	Low	Pending
L-04	Unlock Timestamp Within Game Time	Unexpected Behavior	Low	Pending
L-05	Deposits May Receive The Same Random	Unexpected Behavior	Low	Pending
L-06	Invalid Deposit Amount Emitted	Logical Error	Low	Pending
L-07	Weth Yields Automatically Get Heated	Documentation	Low	Pending

Findings & Resolutions

ID	Title	Category	Severity	Status
L-08	Unnecessary Modulo	Optimization	Low	Pending
L-09	Lacking Configuration Validations	Validation	Low	Pending

H-01 | Invalid Portion Burned

Category	Severity	Location	Status
Logical Error	High	AfterBurner.sol: 174	Resolved

Description

In the reheat function the `reserveSize` is computed with a denominator of 100e18, however the `minPortion` and `maxPortion` are assigned to as .05 ether and .15 ether respectively in the constructor.

The comment on line 150 indicates that the portion ought to be 15% rather than 0.15%, therefore the portion is a factor of 100x smaller than it ought to be.

Recommendation

Use a denominator of 1e18 rather than 100e18.

Resolution

Baseline Team: Resolved.

H-02 | YesArena Block Stuffing Attack

Category	Severity	Location	Status
Block Stuffing	High	YesArena.sol	Pending

Description

A malicious actor may significantly increase their odds of winning the YesArena at the end of the game time by using two addresses to ensure they control both the winner and hotPotato addresses and then submitting many transactions to stuff Blast blocks for the next 2 minutes.

For example:
• Bob calls deposit with address A, A is now the hot potato
• Bob calls deposit with address B, A is now the winner & B is the hot potato
• Bob submits many gas waster transactions to stuff the next 2 minutes of Blast blocks until he is the winner

This can significantly reduce the chances that other actors have at getting a deposit call recorded before the 2 minutes is over.

Recommendation

Be aware of this risk, consider adding more time to the game for every deposit to make it more costly to block stuff the chain to improve winning odds.

Resolution

Baseline Team: Pending.

M-01 | claimed Value Not Assigned

Category	Severity	Location	Status
Logical Error	Medium	YesArena.sol: 80	Pending

Description

In the `claim` function the `claimed` boolean is not assigned to `true`, therefore the system never indicates that the claim can occur.

As a result arbitrary users can trigger multiple transfers to the winner and emit the `Claim` event several times.

Recommendation

Assign the `claimed` boolean to true in the `claim` function.

Resolution

Baseline Team: Pending.

M-02 | Reheat Sniping

Category	Severity	Location	Status
Sandwhich Attack	Medium	Afterburner.sol	Pending

Description

The reheat function will buy YES and loop it when the random roll hits. However the random value is based upon the the block.timestamp and block.prevrandao which are both deterministically available at the block in which the reheat transaction is recorded.

In environments where front-running is possible, a malicious actor may create a contract which buys YES and reverts if the block.timestamp and block.prevrandao would not fulfill the random requirements.

This way the attacker can detect owner transactions to reheat and frontrun them in the same block to buy YES right before the price increases as a result of the reheat. The attacker can then back run the reheat and sell their YES tokens if a reheat looping was performed for a risk free immediate profit.

Recommendation

This is not an immediate concern on the Blast L2 network, however be sure to consider this risk before deploying to a network with high MEV activity.

Resolution

Baseline Team: Pending.

M-03 | Blast Yields Are Not Configured For YesArena

Category	Severity	Location	Status
Configuration	Medium	YesArena.sol	Pending

Description

In the YesArena contract there is no configuration for gas yields, however the YesArena contract is likely to accrue a nontrivial gas expenditure during the game.

Recommendation

Consider implementing appropriate configurations and functions to claim the gas yields that would accrue for the YesArena contract.

Resolution

Baseline Team: Pending.

M-04 | Lacking _buy Slippage Protection

Category	Severity	Location	Status
Sandwhich Attack	Medium	Afterburner.sol: 209	Pending

Description

In the `_buy` function there is no slippage protection configured in the swap call. This allows malicious actors to sandwich the `reheat` transaction's swap and extract value from the system.

Recommendation

The system is currently deployed on Blast which does not have a public mempool, so frontrunning sandwich vectors are not an immediate concern.

However upon deploying to new chains, carefully consider this risk and implement the necessary swap protections to mitigate the sandwich attack vector.

Resolution

Baseline Team: Pending.

L-01 | Lack Of Upgradeability Controls

Category	Severity	Location	Status
Suggestion	Low	AfterBurner.sol	Pending

Description

The `mm` and `cf` addresses are declared `immutable` in the `AfterBurner` contract, however In the event that the `MarketMaking` or `CreditFacility` contracts are upgraded, a new `AfterBurner` contract would need to be deployed. This may become unwieldy and incur the team unnecessary deploy expenses over time.

Recommendation

Consider implementing functions to update the cf and mm addresses, and be sure that the owner address is a multi-sig. Otherwise if trust of the owner is a concern, do not add these functions and be aware that the AfterBurner should be re-deployed with funds ported over in the event of a `MarketMaking` or `CreditFacility` contract upgrade.

Resolution

Baseline Team: Pending.

L-02 | Lacking Rate Validations

Category	Severity	Location	Status
Validation	Low	YesArena.sol: 39	Pending

Description

In the `YesArena` contract constructor there is no validation that the `GROWTH_RATE` is correctly assigned to a value greater than `1e18`. If the `GROWTH_RATE` value is assigned to less than `1e18` it will result in a smaller deposit price over time. Similarly, there is no validation requiring the `FEE_RATE` to be a reasonable proportion of `1e18`.

Recommendation

Consider implementing validations such that the `GROWTH_RATE` cannot be assigned to a value less than `1e18` and the `FEE_RATE` cannot be above a certain threshold.

Resolution

Baseline Team: Pending.

L-03 | YesArena References Old AfterBurner

Category	Severity	Location	Status
Configuration	Low	YesArena.sol: 24	Pending

Description

In the `YesArena` contract the `afterburner` address is hardcoded as the existing afterburner contract, which does not include the latest updates.

Recommendation

Consider making the `afterburner` address configurable within the constructor. Otherwise be sure to update this address in the `YesArena` contract before deployment.

Resolution

Baseline Team: Pending.

L-04 | Unlock Timestamp Within Game Time

Category	Severity	Location	Status
Unexpected Behavior	Low	YesArena.sol: 108	Pending

Description

The `UNLOCK_TIMESTAMP` may occur within the `gameTime` period if enough deposits are made, as a result the winner will be able to claim the jackpot immediately after winning.

Recommendation

Consider if this is expected behavior, if it is not then consider altering the unlock time validation such that it validates that a certain amount of time has passed since the end of the game time period.

Resolution

Baseline Team: Pending.

L-05 | Deposits May Receive The Same Random

Category	Severity	Location	Status
Unexpected Behavior	Low	YesArena.sol: 89	Pending

Description

In the deposit function a pseudo random value is generated to be emitted with the Deposited event for the deposit. However this random value is generated based upon values that apply to the entire block, not just the particular transaction being executed.

As a result several deposits within the same block will have the same random value associated with them in the Deposited event.

Recommendation

Consider if this is the expected behavior, otherwise consider seeding the deposit with values that can distinguish each deposit within a single block from each other, such as the depositNumber.

Resolution

Baseline Team: Pending.

L-06 | Invalid Deposit Amount Emitted

Category	Severity	Location	Status
Logical Error	Low	YesArena.sol: 99	Pending

Description

In the deposit function the `Deposited` event emits the `depositPrice` as the amount variable, however this is not the amount which the caller paid as the `depositPrice` was subsequently increased by the growth rate.

Recommendation

Consider caching the `depositPrice` variable at the beginning of the deposit function and emitting this as the amount in the `Deposited` event.

Additionally, use this cached `depositPrice` stack variable to perform the deposit validation, transfers, and new `depositPrice` calculation to save gas in the `deposit` function.

Resolution

Baseline Team: Pending.

L-07 | Weth Yields Automatically Get Heated

Category	Severity	Location	Status
Documentation	Low	Afterburner.sol	Pending

Description

In the `Afterburner` contract any yield for the `weth` reserve assets held in the `Afterburner` will automatically be included in the reheat actions as they will automatically be applied to the `Afterburner` contract balance.

This may be expected, however it is worth pointing out as the protocol may wish to claim these yields instead of having them automatically attributed to each reheat.

Recommendation

Consider if the `weth` yields should be applied to reheat actions, otherwise implement logic in the constructor such that the yield mode is claimable and a trusted address may withdraw these yields for the protocol.

Resolution

Baseline Team: Pending.

L-08 | Unnecessary Modulo

Category	Severity	Location	Status
Optimization	Low	Afterburner.sol: 142	Pending

Description

In the `reheat` function the hit value is determined based on `roll % probabilityDenominator == 69`, however the roll value has already been modded by the `probabilityDenominator` and incremented by 1.

Therefore modding by the `probabilityDenominator` a second time only maps rolls of 100 to 0, and therefore will not affect the odds of a hit.

Recommendation

Remove the second module which occurs on line 142 as it is unnecessary.

Resolution

Baseline Team: Pending.

L-09 | Lacking Configuration Validations

Category	Severity	Location	Status
Validation	Low	Afterburner.sol	Pending

Description

In the `Afterburner` contract there are several owner configuration functions which lack important validations. For example, in the `setSources` function, the sources array should be validated to be within a reasonable length such that the `reheat` function cannot be accidentally or maliciously DoS'd due to an extremely long `sources` array.

In the `setPortionBounds` and `setDelayBounds` functions there is no validation that the configured min bound is less than the max bound. And the `setProbabilityDenominator` does not validate that the denominator is nonzero.

Recommendation

Consider implementing the suggested validations to protect against accidental assignments or owner compromises.

Resolution

Baseline Team: Pending.

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