Augur Core Audit

OPENZEPPELIN SECURITY | MARCH 12, 2018

Security Audits

The Augur team asked us to review and audit their Augur Core project. We looked at the code and now publish our results.

The audited project can be found in the AugurProject/augur-core GitHub repository. The version used for this report is commit `45e1afb7eb1a895d923c97fe01e068c772c583ef`.

***Update**: the Augur team implemented some of our recommendations and added new features, after which we performed a second audit round. The commit for the final audit is `3b5a63d372d205a0214e3061293d5bca0fd5636a`.*

***Update 2**: the Augur team implemented our additional recommendations after the second audit. The final commit containing all the fixes is `7f3c79a5dd471a98df8f66a640902e063f15f796`.*

The full report can be found here, and a list summary of the issues ordered by severity can be found next.

Critical Severity

- Use safe math (new)
- An attacker can manipulate the tentative winning outcome in case a fork
- An attacker can prevent forking-market traders from claiming their fees
- Markets can be migrated after finalization
- Markets are not sanity-checked in trading module

- Order info is repeated as arguments when cancelling an order
- It may not be possible to stake tokens on an invalid outcome
- Markets ether balance can be stolen by the first reporter
- All reporting fees can be frozen by a Market creator
- A market owner can block the Participation token purchase

High Severity

- Extractable functionality is not necessary and error prone
- Non-potential-winning dispute crowdsourcers can redeem their REP tokens
- Market number of ticks can be zero
- Self-reference in market nudging mechanism
- Tight coupling between contracts
- Anyone can trigger Augur events
- Cancelling an order with share tokens in escrow will fail
- Markets can be created with malicious Cash tokens
- Shareholders fees can be frozen by a malicious market creator
- Spender contracts cannot be re-approved if updated

Medium Severity

- Favor pull payments over push payments (new)
- Integer index types are unnecessarily small
- Unbound iteration in arrays (new)
- Unbound iteration in arrays
- Users are allowed to place orders for a market independently of their state
- Unclear relation between MIN_ORDER_VALUE and MINIMUM_GAS_NEEDED
- Reentrancy risk in FillOrder
- Markets can be initially reported in a locked universe
- Forking market can be migrated
- Fork values for child universes must be manually updated
- Trading contracts upgradeability may become useless
- Controller does not guarantee that dev mode cannot be turned on again

- Markets can be created in a locked universe
- Eventually it will not be possible to produce further forks
- Spender contracts cannot be re-approved if updated
- Naming issues
- Repeated code for factory contracts
- Unused boolean return values
- Unsolved TODO comments
- Instances of Map contract left in blockchain storage
- Unused Set library
- Inconsistent usage of getter functions and state variables
- Use a standard toolchain for building contracts
- No assertions for detecting broken invariants
- Install OpenZeppelin via NPM
- OpenZeppelin standard tokens were modified
- Outdated OpenZeppelin's contracts
- Outdated documentation
- Orders are vulnerable to front-running
- Basic token implementation allows transfers to the zero address
- Lack of Report abstraction
- Universe open interest is not decremented in bad times
- Markets can fork into more than N+1 universes, N being the number of outcomes (new)
- Markets may fork in more than N+1 universes, N being the number of outcomes
- When a market forks, stake tokens and disputes of other markets are reset
- Unchecked token transfers and approvals
- ShareToken is unnecessarily whitelisted
- Use safe math
- Remove unused code
- The Trade logic treats a lack of gas as a complete order fill
- Market creators may not be able to collect their corresponding fees
- Delegator memory allocation not working for arguments larger than 32 bytes
- Delegator not working for return data greater than 32 bytes

- Repeated code for factory contracts
- Unused boolean return values
- Unsolved TODO comments
- Instances of Map contract left in blockchain storage
- Unused Set library
- Inconsistent usage of getter functions and state variables
- Use a standard toolchain for building contracts
- No assertions for detecting broken invariants
- Install OpenZeppelin via NPM
- OpenZeppelin standard tokens were modified
- Outdated OpenZeppelin's contracts
- Outdated documentation
- Orders are vulnerable to front-running
- Basic token implementation allows transfers to the zero address
- Lack of Report abstraction
- Universe open interest is not decremented in bad times
- Markets can fork into more than N+1 universes, N being the number of outcomes (new)
- Markets may fork in more than N+1 universes, N being the number of outcomes
- When a market forks, stake tokens and disputes of other markets are reset
- Unchecked token transfers and approvals
- ShareToken is unnecessarily whitelisted
- Use safe math
- Remove unused code
- The Trade logic treats a lack of gas as a complete order fill
- Market creators may not be able to collect their corresponding fees
- Delegator memory allocation not working for arguments larger than 32 bytes
- Delegator not working for return data greater than 32 bytes

Conclusion

Thirteen critical and ten high severity issues were found, along with recommendations on how to fix them. Additionally, some medium and lower severity issues were found and explained. Some changes were proposed to follow best practices and reduce the potential attack surface.

security, check out our thoughts here.

Related Posts

Beefy Zap Audit

BeefyZapRouter serves as a versatile intermediary designed to execute users' orders through routes...

Security Audits

OpenBrush Contracts Library Security Review

OpenBrush is an open-source smart contract library written in the Rust programming language and the...

Security Audits

Linea Bridge Audit

Linea is a ZK-rollup deployed on top of Ethereum. It is designed to be EVM-compatible and aims to...

Security Audits

OpenZeppelin

Company

About us
Jobs
Blog

Contracts Library

Docs

Privacy | Terms of Use