GoatX Security Audit

Report Version 1.0

December 27, 2024

Conducted by **Hunter Security**

Table of Contents

1 About Hunter Security

Hunter Security is an industry-leading smart contract security auditing firm. Having conducted over 100 security audits protecting over $1B of TVL, our team always strives to deliver top-notch security services to the best DeFi protocols. For security audit inquiries, you can reach out on Telegram or Twitter at *@georgehntr*.

2 Disclaimer

Audits are a time-, resource-, and expertise-bound effort where trained experts evaluate smart contracts using a combination of automated and manual techniques to identify as many vulnerabilities as possible. Audits can reveal the presence of vulnerabilities, but cannot guarantee their absence.

3 Risk classification

Severity	Impact: High	Impact: Medium	Impact: Low
Likelihood: High	High	High	Medium
Likelihood: Medium	High	Medium	Low
Likelihood: Low	Medium	Low	Low

3.1 Impact

- **High** - leads to a significant loss of assets in the protocol or significantly harms a group of users.
- **Medium** - involves a small loss of funds or affects a core functionality of the protocol.
- **Low** - encompasses any unexpected behavior that is non-critical.

3.2 Likelihood

- **High** - a direct attack vector; the cost is relatively low compared to the potential loss of funds.
- **Medium** - only a conditionally incentivized attack vector, with a moderate likelihood.
- **Low** - involves too many or unlikely assumptions; offers little to no incentive.

3.3 Actions required by severity level

- **High** - client **must** fix the issue.
- **Medium** - client **should** fix the issue.
- **Low** - client **could** fix the issue.

4 Executive summary

Overview

Project Name	GoatX
Repository	https://github.com/De-centraX/goatx-contracts
Commit hash	065dd4ac83e4720a304921010a7b4e4d7bb0a0ef
Resolution	6e5d65fef79ae51b616fd3d36c0f81b8342134df
Methods	Manual review & testing

Scope

src/actions/SwapActions.sol
src/Auction.sol
src/AuctionBuy.sol
src/BuyAndBurn.sol
src/GoatFeed.sol
src/GoatX.sol
src/Minting.sol

Issues Found

High risk	0
Medium risk	0
Low risk	0
Informational	7

5 Findings

5.1 Informational

5.1.1 Typographical mistakes, non-critical issues and code-style suggestions

Severity: Informational

Context: src/*

Description: The contracts contains one or more typographical mistakes, non-critical issues and code-style suggestions. In an effort to keep the report size reasonable, we enumerate these below:

1. Consider inheriting OpenZeppelin's Ownable2Step rather than Ownable.
2. Multiple storage variables do not have explicit visibility specified.
3. The *buyNBurn* method would not work until the initial LP has been collected.
4. Consider reverting in case *oldestObservation* < *secondsAgo* in *getTwapAmount* to prevent price manipulation.
5. The *fluxBnb* variables is never used.
6. Remove unnecessary early *uint8* cast in *_getCycleAt* to prevent silent overflow.
7. If *getRatioForCycle* is used externally it may cause return misleading result in case of underflow. Consider either removing the *unchecked* statement or making internal.

Recommendation: Consider fixing the above typographical mistakes, non-critical issues and code-style suggestions.

Resolution: Partially resolved.