Hyper Security Audit

Report Version 1.0

March 7, 2024

Conducted by:

George Hunter, Independent Security Researcher

Table of Contents

1 About George Hunter

George Hunter is a proficient and reputable independent smart contract security researcher with over 50 solo and team security engagements contributing to the security of numerous smart contract protocols in the past 2 years. Previously held roles include Lead Smart Contract Auditor at Paladin Blockchain Security and Smart Contract Engineer at Nexo. He has audited smart contracts for clients such as LayerZero, Euler, TraderJoe, Maverick, Ambire, and other leading protocols.

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

George Hunter was engaged by Hyper to review the Hyper smart contract protocol (a fork of TitanX) during the period from February 26, 2024, to February 29, 2024.

Overview

Repository	https://github.com/jakesharpe777/ttx_hyper_new_private
Commit hash	bced910d9369fa936892461922d0dbeb54626cfa
Resolution	71974f890134895e6de7455ae7624ad0eccebaae

Repository	https://github.com/jakesharpe777/ttx_hyper_new_buyandburn_private
Commit hash	146987362b06b45405acf9e1274801987f39b7e0
Resolution	146987362b06b45405acf9e1274801987f39b7e0

Timeline

-	February 26, 2024	Audit kick-off
v0.1	March 3, 2024	Preliminary report
v1.0	March 7, 2024	Mitigation review

Scope

contracts/BurnInfo.sol
contracts/GlobalInfo.sol
contracts/Hyper.sol
contracts/MintInfo.sol
contracts/OwnerInfo.sol
contracts/StakeInfo.sol
libs/*

contracts/BuyAndBurnHyper.sol
libs/*

Issues Found

High risk	0
Medium risk	0
Low risk	0
Informational	1

5 Findings

5.1 Informational

5.1.1 Typographical mistakes and code style suggestions

Severity: *Informational*

Context: Hyper.sol#L338, MintInfo.sol#L320-L322

Description: The contracts contains one or more typographical issue(s). In an effort to keep the report size reasonable, we enumerate these below:

1. A boolean type can be used instead of the newly introduced `BuyAndBurnCAStatus`.
2. The zero-address check in `setBuyAndBurnContractAddress` is now redundant due to the calls performed on the followed lines.
3. The following checks are redundant (gas saving):

```
if (balanceOf(msg.sender) < amount) revert Hyper_InsufficientBalance();

if (IERC20(TITANX_CA).balanceOf(msg.sender) < amount) revert
    Hyper_InsufficientBalance();
```

4. The following checks in `MintInfo` can be rewritten to `if (mint.status != MintStatus.Active)revert` for simplicity and better readability/auditability:

```
if (mint.status == MintStatus.CLAIMED) revert Hyper_MintHasClaimed();
if (mint.status == MintStatus.BURNED) revert Hyper_MintHasBurned();
if (mint.status == MintStatus.EARLYENDED) revert Hyper_MintHasEnded();
```

Recommendation: Consider implementing the aforementioned suggestions.

Resolution: Resolved.