HLX Security Audit

Report Version 1.0

March 12, 2025

Conducted by **Hunter Security**

Table of Contents

1 About Hunter Security

Hunter Security is an industry-leading smart contract security company. Having conducted over 100 security audits protecting over $1B of TVL, our team delivers 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 Findings

4.1 Medium

4.1.1 Broken the staking functionality

Severity: Medium

Description: The new HLXBuyBurnV2 contract does not implement a *getCurrentTitanPrice* function.

This would cause a temporary DoS of the HLX staking function due to the following line in *startStake*:

```
? IBuynBurn(s_buyAndBurnAddress).getCurrentTitanPrice()
```

Recommendation: Consider implementing a *getCurrentTitanPrice* function.

Resolution: Resolved.

4.2 Informational

4.2.1 Typographical mistakes and non-critical issues

Severity: Informational

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

1. Consider using a TWAP look back period (*secondsAgo*) of at least 10-15 minutes
2. *capPerSwapEth* and *capPerSwapBuyBurn* should be of *uint128* due to the downcast in *_twapCheck*
3. The *ZeroAddress* and *TWAP* custom errors are defined but not used
4. WETH is never actually the caller in *receive* in HLXBuyBurnV2
5. The admin could set address that revert on *receive* through *setHeliosBuyBurnAddress* and *setHeliosTreasuryAddress* to block the *distributeRewards* function
6. Consider whether *lastStakeTime* and *lastSwapTime* should be initialized in the constructor
7. The *HELIOS_STAKE* constant refers to the HLX contract address
8. *totalHlxBurned* would not always be correct - the BuyAndBurn contract might be different in HLX

Recommendation: Consider fixing the above typographical mistakes and non-critical issues.

Resolution: Partially resolved.