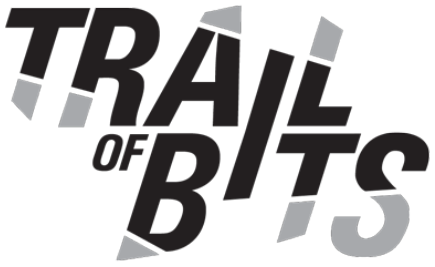

Hegic

Security Assessment Summary

April 17, 2020

Prepared For:
Molly Wintermute | *Hegic*
molly.wintermute@protonmail.com

Prepared By:
Josselin Feist | *Trail of Bits*
josselin@trailofbits.com

Assessment Summary

From April 8 to April 10, 2020, Trail of Bits performed an assessment of the Hegic smart contracts with one engineer, and reported eleven issues ranging from high to informational severity. On April 15, Trail of Bits reviewed the fixes to the reported vulnerabilities.

Md5 hashes of the reviewed files :

- `HegicOptions.sol: 5df703d69a65941a4ea388c0659dafc1`
- `HegicETHPool.sol: 7900140c6393ad43ba343485cec42961`
- `HegicERCPool.sol: 52b51acceec4b615640fb078a095f6a7`
- `HegicPutOptions.sol: 35cf35d69d26a40fdf6f444123a13acf`
- `HegicCallOptions.sol: 39ae6d815e121f8b6ad2b7b7ff354c3c`
- `Interfaces.sol: 36903e242bbd83559a2d9b382512f326`

Throughout this assessment, we sought to answer various questions about the security of the contracts. We focused on flaws that would allow an attacker to:

- Steal assets from a pool
- Create options with a strike price cheaper than expected
- Create options for free

Most of the issues found are related to arithmetic, including:

- Attackers can drain funds if the pool-token's supply is lower than the asset's supply
- A strike amount can be zero if the Ether (ETH) price is under $1
- Malicious pool parameters allow minting of zero tokens when liquidity is added

Several issues would have allowed a malicious contract owner to harm the users, including:

- Stealing option assets through the collection of the fees
- Trapping funds in the option contract, preventing liquidity providers from withdrawing
- Creating options for free

Additionally, we found that the pool had incorrect bookkeeping when adding or removing assets, and did not account for the assets present in the option contract. As a result, an attacker could steal the pool's assets.

Hegic fixed the reported issues after they were reported.

Trail of Bits recommended verifying the contracts' invariants using symbolic execution and fuzzing due to the discovered number of arithmetic issues and time constraints that did not permit in-depth arithmetic verification.

Additionally, Trail of Bits made the following recommendations:

- Use crytic.io/ for future development. Two issues were found using the platform.
- Evaluate and document the owner privileges.
- Verify and document asset bookkeeping across the different contracts.
- Evaluate and document the arbitrage opportunities of the system.
- Recommend users call the `provide` and `withdraw` functions with asset guarantees.